View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Infoscience - École polytechnique fédérale de Lausanne

ENERGY EFFICIENT MEDIUM ACCESS CONTROL FOR WIRELESS SENSOR NETWORKS

THÈSE N^{O} 3285 (2005)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Institut d'informatique fondamentale

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDERALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Amre EL-HOIYDI

ingénieur électricien diplômé EPF
de nationalité suisse et originaire de Delémont (JU)

acceptée sur proposition du jury:

Prof. J.-D. Decotignie, directeur de thèse
Prof. J.-C. Grégoire, rapporteur
Dr A. Kaelin, rapporteur
Prof. A. Schiper, rapporteur

Lausanne, EPFL
2005

Acknowledgement

First of all, I would like to thank my advisor Prof. Jean-Dominique Decotignie, to have given me the opportunity to pursue a PhD in his group. His far-seeing guidance directed my research towards an area that proved to be very interesting. I am grateful for his high availability and for the support and trust he gave me.

Most of the work presented within this dissertation was made within the WiseNET project at CSEM. I would like to thank all the participants of the WiseNET project. In particular, I would like to thank Erwan Le Roux for having spent many hours explaining me the workings of the WiseNET system-on-chip and to have guided my work towards the preamble sampling technique, which proved to be a fruitful direction. Special thanks go also to Thierry Melly, Patrick Volet and Christian Enz.

Working in the Real-Time Software and Networking group at CSEM was a great pleasure. I am thankful to my colleagues in *secteur* 241 for there friendship and helpfulness. I would like to thank particularly Jean Hernandez to have introduced me to the field of low power communication protocols.

I am grateful to Prof. Jean-Charles Grégoire, Dr. August Kaelin and Prof. André Schiper for having accepted to be part of my jury.

The work presented in this dissertation was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

This dissertation is dedicated to my son Felix, my wife Caroline and my parents Madeleine and Ahmed.

Amre El-Hoiydi
Neuchâtel, July 2005

Abstract

A wireless sensor network designates a system composed of numerous sensor nodes distributed over an area in order to collect information. The sensor nodes communicate wirelessly with each other in order to self-organize into a multi-hop network, collaborate in the sensing activity and forward the acquired information towards one or more users of the information. Applications of sensor networks are numerous, ranging from environmental monitoring, home and building automation to industrial control.

Since sensor nodes are expected to be deployed in large numbers, they must be inexpensive. Communication between sensor nodes should be wireless in order to minimize the deployment cost. The lifetime of sensor nodes must be long for minimal maintenance cost. The most important consequence of the low cost and long lifetime requirements is the need for low power consumption. With today's technology, wireless communication hardware consumes so much power that it is not acceptable to keep the wireless communication interface constantly in operation. As a result, it is required to use a communication protocol with which sensor nodes are able to communicate keeping the communication interface turned-off most of the time.

The subject of this dissertation is the design of medium access control protocols permitting to reach a very low power consumption when communicating at a low average throughput in multi-hop wireless sensor networks.

In a first part, the performance of a scheduled protocol (time division multiple access, TDMA) is compared to the one of a contention protocol (non-persistent carrier sensing multiple access with preamble sampling, NP-CSMA-PS). The preamble sampling technique is a scheme that avoids constant listening to an idle medium. This thesis presents a low power contention protocol obtained through the combination of preamble sampling with non-persistent carrier sensing multiple access. The analysis of the strengths and weaknesses of TDMA and NP-CSMA-PS led us to propose a solution that exploits TDMA for the transport of frequent periodic data traffic and NP-CSMA-PS for the transport of sporadic signalling traffic required to setup the TDMA schedule.

The second part of this thesis describes the WiseMAC protocol. This protocol is a further enhancement of CSMA with preamble sampling that proved to provide both a low power consumption in low traffic conditions and a high energy efficiency in high traffic conditions. It is shown that this protocol can provide either a power consumption or a latency several times lower that what is provided by previously proposed protocols. The WiseMAC protocol was initially designed for multi-hop wireless sensor networks. A comparison with power saving protocols designed specifically for the downlink of infrastructure wireless networks shows that it is also of interest in such cases. An implementation of the WiseMAC protocol has permitted to validate experimentally the proposed concepts and the presented analysis.

Version abrégée

Un réseau de capteurs sans fil est un système composé de nombreux capteurs distribués sur une zone pour collecter de l'information. Les capteurs communiquent entre eux par ondes radio pour auto-organiser la formation du réseau, pour collaborer dans les activités de mesure et pour acheminer l'information collectée vers un ou plusieurs utilisateurs de cette information. Les applications de réseaux de capteurs sans fil sont nombreuses. Elles comprennent notamment la surveillance de l'environnement naturel ou construit (agriculture, génie civil, etc) et l'automatisation dans les bâtiments (sécurité, contrôle de la ventilation, du chauffage, etc).

Pour pouvoir être déployés en grand nombre, les capteurs doivent être bon marché. La communication entre les capteurs doit se faire sans fil pour permettre un bas coût d'installation. La durée de vie d'un capteur doit être longue pour minimiser les coûts de maintenance. La conséquence des ces besoins est que leur consommation en énergie doit être faible. Avec la technologie actuelle, le matériel permettant une communication par ondes radio consomme une quantité d'énergie telle qu'un fonctionnement permanent est inacceptable. Il est donc nécessaire d'utiliser un protocole de communication permettant aux capteurs de communiquer tout en gardant leur interface radio éteinte la plupart du temps.

Cette dissertation a pour sujet la conception de protocoles de gestion d'accès permettant une très faible consommation d'énergie pour des communications à faible débit dans des réseaux de capteurs distribués.

Dans la première partie, les performances d'un protocole utilisant une organisation temporelle (multiplexage temporel, ou time division multiple access TDMA) sont comparées à celles d'un protocole utilisant une méthode d'accès par compétition (méthode d'accès multiple avec écoute de porteuse, sans persistance, et avec échantillonnage de préambule, ou non-persistent carrier sensing multiple access with preamble sampling NP-CSMA-PS). La technique de l'échantillonnage de préambule permet d'éviter l'écoute permanente d'un canal libre. Cette thèse présente un protocole à basse consommation d'énergie utilisant une méthode d'accès par compétition, obtenue en combinant la méthode d'accès multiple avec écoute de porteuse avec la technique de l'échantillonnage de préambule. L'analyse des avantages et faiblesses des protocoles TDMA et NP-CSMA-PS a conduit à proposer une solution qui exploite à la fois le multiplexage temporel pour le transport du trafic périodique et fréquent des informations collectées, et la méthode d'accès par compétition pour le transport du trafic sporadique utile à la mise en place du multiplexage temporel.

La deuxième partie de cette thèse décrit le protocole WiseMAC. Ce protocole est une version améliorée de NP-CSMA-PS qui permet d'obtenir avec un seul protocole une très basse consommation pour le transport de trafic sporadique et une haute efficacité énergétique en cas de grand trafic. Il est démontré que ce protocole permet d'obtenir une consommation énergétique ou une latence plusieurs fois inférieur à ce que permettent d'obtenir les protocoles proposés auparavant.

Ce protocole a été conçu initialement pour des réseaux à sauts multiples. Une comparaison avec les protocoles conçus spécifiquement pour le lien descendant de réseaux sans fil à infrastructure a démontré que WiseMAC est aussi intéressant dans ces cas. Une implémentation du protocole WiseMAC a permit de valider expérimentalement les concepts proposés et l'analyse présentée.

Contents

List of Figures 13

List of Tables 17

1 Introduction 1
- 1.1 Topologies 1
- 1.2 Requirements 3
- 1.3 Low power design 4
- 1.4 Energy efficient communication 4
  - 1.4.1 Physical layer 4
  - 1.4.2 Data link layer 5
  - 1.4.3 Network layers 5
  - 1.4.4 Transport, session, and presentation layers 6
- 1.5 Problem statement 6
- 1.6 Contributions 6
- 1.7 Thesis organization 7

2 Energy Efficient Medium Access Control - State of the Art 9
- 2.1 Medium access control 9
- 2.2 Sources of energy waste at MAC layer 10
- 2.3 Power saving schemes at MAC layer 11
  - 2.3.1 Wireless Local Area Networks (WLANs) 11
    - 2.3.1.1 IEEE 802.11 infrastructure mode 11
    - 2.3.1.2 Hiperlan 2 12
  - 2.3.2 Mobile Ad-Hoc Networks (MANETs) 13
    - 2.3.2.1 IEEE 802.11 ad-hoc mode 13
    - 2.3.2.2 Hiperlan 1 14
  - 2.3.3 Paging systems 14
    - 2.3.3.1 POCSAG 14
    - 2.3.3.2 FLEX and ERMES 15
  - 2.3.4 Wireless Personal Area Networks (WPANs) 15
    - 2.3.4.1 Bluetooth 15
    - 2.3.4.2 IEEE 802.15.4 16
  - 2.3.5 Wireless sensor networks 18
    - 2.3.5.1 Scheduled MAC Protocols 19
    - 2.3.5.2 Unscheduled MAC Protocols 21

2.4 Summary 23

3 **Battery and Transceiver Models** **25**
3.1 Introduction 25
3.2 Battery model 25
3.3 Radio transceiver model 26
3.3.1 Model parameters 26
3.3.1.1 Power consumption and transition delays 26
3.3.1.2 Other parameters 28
3.3.2 WiseNET SoC model 29
3.4 Conclusion 30

4 **Spatial TDMA and Non-Persistent CSMA with Preamble Sampling** **33**
4.1 Introduction 33
4.2 Spatial TDMA 33
4.2.1 Scenario 33
4.2.2 Required synchronization interval 34
4.2.3 Power consumption 35
4.3 Non-persistent CSMA with preamble sampling 37
4.3.1 Preamble sampling 37
4.3.2 Carrier sensing protocols 38
4.3.3 Non-persistent CSMA with preamble sampling 40
4.3.3.1 Throughput 41
4.3.3.2 Delay 42
4.3.3.3 Power consumption 43
4.3.4 Regular and genie aided NP-CSMA 44
4.3.5 Performance evaluation 45
4.3.6 Mitigating overhearing 48
4.4 Comparing and combining S-TDMA and NP-CSMA-PS 49

5 **WiseMAC for Multihop Wireless Sensor Networks** **53**
5.1 Introduction 53
5.2 Protocol description 53
5.2.1 Overview 53
5.2.2 Minimized wake-up preamble 54
5.2.3 Medium reservation 55
5.2.4 Random backoff 59
5.2.5 Overhearing mitigation 60
5.2.6 ”More” bit 64
5.2.7 Inter-frame spaces 64
5.2.8 Receive and carrier sense thresholds 65
5.2.9 Sampling period 68
5.3 Performance analysis 69
5.3.1 Reference protocols 70
5.3.1.1 Ideal protocol 70
5.3.1.2 S-MAC 71

5.3.1.3 T-MAC 71
5.3.1.4 CSMA/CA 72
5.3.2 Theoretical power consumption 72
5.3.2.1 Ideal protocol 73
5.3.2.2 S-MAC 73
5.3.2.3 WiseMAC 74
5.3.3 Simulation in a lattice network 76
5.3.3.1 Topology 76
5.3.3.2 Traffic 76
5.3.3.3 Receive, interference and carrier sense ranges 77
5.3.3.4 Hop delay 78
5.3.3.5 Power consumption 79
5.3.3.6 Throughput 81
5.3.3.7 Lifetime 81
5.3.3.8 Energy efficiency 82
5.3.4 Simulation in a random network 83
5.3.4.1 Topology and traffic 83
5.3.4.2 Power consumption and delay 84
5.4 Sensitivity analysis 85
5.4.1 Impact of the sampling period 85
5.4.2 Impact of the different schemes used in WiseMAC 86
5.4.3 Impact of external interferences 87
5.4.4 Importance of the transceiver parameters 88
5.4.5 Impact of the quartz frequency tolerance 88
5.4.6 Impact of the battery model 89
5.5 Conclusion 89

6 Downlink of an Infrastructure Wireless Sensor Network 91
6.1 Introduction 91
6.1.1 Problem statement 91
6.1.2 Traffic model 92
6.1.3 Chapter outline 92
6.2 Low power downlink MAC protocols 93
6.2.1 Ideal protocol 93
6.2.2 WiseMAC 94
6.2.3 Periodic Terminal Initiated Polling - PTIP 94
6.2.4 IEEE 802.11/802.15.4 Power Save Mode - PSM 95
6.2.5 Adaptability of the wake-up period 97
6.3 Power consumption 97
6.3.1 Power consumption of the ideal protocol 97
6.3.2 Power consumption of WiseMAC 98
6.3.3 Power consumption of PTIP 98
6.3.4 Power consumption of PSM 100
6.4 Transmission delay 100
6.4.1 Delay with the ideal protocol 100
6.4.2 Delay with WiseMAC 101

6.4.3 Delay with PTIP . . . 102
6.4.4 Delay with PSM . . . 103
6.5 Performance comparison . . . 103
6.6 Sensitivity analysis . . . 106
6.6.1 Sensitivity to traffic . . . 106
6.6.2 Scalability . . . 106
6.6.3 Impact of the data frame repetition in the WiseMAC preamble . . . 108
6.6.4 Impact of the packet size . . . 110
6.6.5 Impact of the quartz frequency tolerance . . . 111
6.6.6 Impact of the TX/RX power consumption ratio . . . 111
6.6.7 Impact of the bit rate . . . 112
6.7 Conclusion . . . 114

7 Experimentation 117
7.1 Introduction . . . 117
7.2 Hardware platform . . . 117
7.2.1 Microcontroller . . . 117
7.2.2 Radio transceiver . . . 118
7.2.3 Development and demonstration boards . . . 119
7.3 Software architecture . . . 119
7.4 Measurements . . . 120
7.4.1 Time-keeping base accuracy . . . 120
7.4.2 Static current consumption . . . 120
7.4.3 Dynamic current consumption . . . 121
7.4.4 Energy consumption of sampling . . . 123
7.4.5 Minimization of the wake-up preamble length . . . 123
7.4.6 Multi-hop transmission . . . 125
7.4.7 Average power consumption and transmission delay . . . 126
7.5 Conclusion . . . 128

8 Conclusion 129

A Interference Between Bluetooth Piconets 131
A.1 Introduction . . . 131
A.2 Model . . . 131
A.3 Packet Error Rate . . . 132
A.4 Aggregated Throughput . . . 136
A.5 Simulation Model in OPNET . . . 136
A.6 Simulations Results . . . 139
A.7 Conclusion . . . 141

B Simulation Model 143
B.1 Simulation plateform . . . 143
B.2 Interference and radio layer simulation model . . . 143

Bibliography 147

List of Figures

1.1 Sensor network topologies. 2

2.1 IEEE 802.11 infrastructure network, power saving mode for downlink communication. 12

2.2 Power saving in an IEEE 802.11 ad-hoc network. 13

2.3 POCSAG paging system frame format. 14

2.4 IEEE 802.15.4 beacon-enabled superframe structure. 16

2.5 Downlink data transfer in a beacon-enabled IEEE 802.15.4 network. 16

2.6 Uplink data transfer in a beacon-enabled IEEE 802.15.4 network. 17

2.7 A classification of MAC protocols for wireless sensor networks. 19

2.8 S-MAC. 21

2.9 Piconet. 22

2.10 Operation of the paging channel in STEM. 23

3.1 Typical discharge curve for alkaline/manganese batteries (left) and for lithium/manganese batteries (right). 26

3.2 Lifetime of a sensor node using a single 2.6 Ah alkaline battery as a function of the average consumed power. 27

3.3 Transceiver states, power consumption and transition delays. 27

3.4 Current consumption of the WiseNET SoC during setup phase into receive state. 30

4.1 Spatial TDMA. 34

4.2 TDMA communication with earlier listening for clock drift compensation. 34

4.3 Required synchronization period due to source and destination clock drifts. T is the target transmission time. T_{S1} and T_{S2} are the early and late limits for the start of the transmission by the source. T_D is the target time for listening at the destination. T_{D1} and T_{D2} are the early and late limits for the effective start of the listening phase by the destination. 36

4.4 Power consumption and lifetime with the TDMA protocol, when forwarding 60 bytes packet every L seconds using the WiseNET transceiver. 37

4.5 Preamble Sampling. 37

4.6 Residual vulnerability period with CSMA. 39

4.7 Non-persistent CSMA with preamble sampling. 41

4.8 System model for NP-CSMA analysis. 41

4.9 Initial waiting delay and waiting delay for retransmissions with NP-CSMA. . . . 43

4.10 Performance of non-persistent CSMA with preamble sampling, as compared to classical NP-CSMA and genie aided NP-CSMA. 46
4.11 Power consumption as a function of the interval between successful transmissions. 47
4.12 Approximation without considering collisions. 47
4.13 Percentage of the time spent by the transceiver its different states using NP-CSMA-PS, as a function of the packet inter-arrival. 48
4.14 Comparison between the power consumption of TDMA and NP-CSMA-PS. . . . 50
4.15 Lifetime using Spatial TDMA and NP-CSMA-PS as a function of the data and signalling traffic. 51

5.1 Minimizing the wake-up preamble length. 54
5.2 Clock drift compensation. 55
5.3 Packet format. 55
5.4 Adaptivity of the per-packet overhead to the traffic. 56
5.5 Systematic collision between two nodes transmitting to the same destination at the same sampling time (A) and medium reservation (B). 56
5.6 Probability for a node to capture the medium, defer its transmission or enter in collision, for different number of contenders and as a function of the medium reservation window size. The markers show simulation results. 58
5.7 Computation of the energy consumption of a collision resolution. 58
5.8 Energy consumption per node (upper plot) and number of required contentions per node (lower plot) with a contention resolution between C nodes. 59
5.9 Probabilistic overhearing avoidance. 61
5.10 Repetition of data message within wake-up preamble. 61
5.11 Duration of the overhearing period with WiseMAC, when $T_P < T_D$ (A), $T_D \leq T_P < T_W - T_D$ (B) and $T_W - T_D \leq T_P \leq T_W$ (C). 62
5.12 Average power wasted by a node overhearing transmissions, as a function of the interval between transmissions ($T_W = 50, 200, 500$ ms, $T_D = 19.2$ ms). 63
5.13 Transmission of packet bursts using the ”more” bit. 65
5.14 Extended carrier sensing range for hidden node effect mitigation. 67
5.15 Receiving, interfering and sensing range (upper plot) and receive threshold as a function of the decay index (lower plot). 68
5.16 Effect of a wrong estimation of the decay index. The estimated decay index used to compute the receive threshold is equal to 3.5 in both cases. The effective decay index is equal to 3 on the left plot, and to 4 and the right plot. 69
5.17 Lifetime when sampling the medium with period T_W (no traffic). 69
5.18 Ideal Protocol. 70
5.19 Required duration for the listen interval in S-MAC. 72
5.20 T-MAC inactivity timeout. 73
5.21 Overhearing duration T_O as a function of the preamble duration T_P ($T_W =$ 100 ms, $T_D = 19.2$ ms). 76
5.22 Lattice network topology. 77
5.23 Hop delay as a function of the injected traffic (packets have a length of 60 bytes, $T_W = 100$ ms). 79
5.24 Average power consumption as a function of the injected traffic (packets have a length of 60 bytes, $T_W = 100$ ms). 80

5.25 Average throughput as a function of the injected traffic (packets payloads have a length of 46 bytes, $T_W = 100$ ms). 81
5.26 Lifetime as a function of the traffic when using a single AA alkaline battery leaking 10% of its initial capacity every year. 82
5.27 Energy efficiency. 83
5.28 Random network topology. 84
5.29 Average power consumption. 85
5.30 Average end-to-end delay. 85
5.31 Power consumption (left) and delay (right) with WiseMAC as a function of traffic intensity with different sampling periods T_W =50, 100, 200, 500 ms. 86
5.32 Throughput (top, left), delay (top, right), power consumption (bottom, left) and lifetime (bottom, right) with WiseMAC as a function of traffic intensity without one of the following schemes: Wake-up preamble minimization, extended carrier sensing, more bit, mandatory inter-frame space before transmission. 87
5.33 Power consumption (left) and lifetime (right) with different values for the quartz tolerance: $\theta = 30$, 50 and 100 ppm. 88
5.34 Lifetime as a function of the traffic intensity when using a single AA alkaline battery leaking 3% of its initial capacity every year. 89

6.1 Infrastructure wireless sensor network. 92
6.2 Downlink traffic model. 93
6.3 Ideal protocol. 94
6.4 WiseMAC for the downlink of an infrastructure network. 95
6.5 WiseMAC: Transmission of several packets using the *more* bit. 95
6.6 Periodic Terminal Initiated Polling (PTIP). 96
6.7 Optimized Power Save Mode (PSM). 97
6.8 Power consumption and delay of WiseMAC, PTIP and PSM as a function of the wake-up period T_W ($L = 1000$ s). 104
6.9 Power-delay characteristics of WiseMAC, PTIP and PSM ($L = 1000$ s). 105
6.10 Power consumption as a function of the inter-arrival L ($T_W = 1$ s). 107
6.11 Power consumption as a function of the number of sensor nodes N, for different values of L ($T_W = 1$ s). 108
6.12 Power consumption as a function of the inter-arrival L when data frames are not repeated in the wake-up preamble ($T_W = 1$ s). 109
6.13 Power consumption as a function of the number of sensor nodes N, for different values of L ($T_W = 1$ s). 110
6.14 Power delay characteristics of WiseMAC, PTIP and PSM for $L = 1000$ s and $\theta = 10, 30, 100$ ppm. A larger power consumption corresponds to a larger θ. . . . 111
6.15 Power consumption of WiseMAC, PTIP and PSM for $L = 1000$ s as a function of the $\overline{P}_T/\overline{P}_R$ ratio. The chosen wake-up period is $T_W = 5$ s to remain in the validity domain of PTIP. 112
6.16 Power-delay characteristics of WiseMAC, PTIP and PSM when using a 11 Mbps Lucent Orinoco IEEE 802.11 transceiver (left) and a 250 kbps Chipcon CC2420 802.15.4 transceiver (right). The inter-arrival time is $L = 1000$ s. 113
6.17 Power consumption of a Lucent Orinoco IEEE 802.11 PC Card during the reception of a beacon. 114

7.1 Simplified hardware architecture of a XE88LC06A microcontroller. 118
7.2 EV108 Development board with XM1203 radio module. 119
7.3 WiseNode: A miniaturized low power wireless sensor node based on the XE1203 radio and the XE88LC06A micro-controller. 119
7.4 Single (left) and dual (right) controller solution. 120
7.5 Code size in words of the different software components. 120
7.6 Current consumption of a XE1203F radio driven by a XE88LC06A micro-controller when traveling clock-wise (left, doze-receive-transmit-doze) and anti-clockwise (right, doze-transmit-receive-doze) in the transceiver state machine. 122
7.7 Current consumed by the XE1203 radio transceiver (thick line) and total consumed current (XE1203 and XE88LC06A) when sampling the medium. 123
7.8 Transmission with wake-up preamble of the length of the sampling period (left) and transmission with a wake-up preamble of minimized size (right) ($T_W = 250$ ms).124
7.9 Detailed view of a transmission with a wake-up preamble of minimized size. . . . 125
7.10 Multi-hop transmission of a packet ($T_W = 250$ ms). 126
7.11 Topology used for average power consumption and average delay measurements. 127
7.12 Average power consumption, computed from the measured time spent with the radio in receive and transmit states ($T_W = 250$ ms). 127
7.13 Average hop transmission delay measured in a circular network composed of 5 nodes ($T_W = 250$ ms). 127

A.1 Interference between piconets. 132
A.2 Single or double exposition to interference. 133
A.3 Packet error rate suffered by one piconet because of the interference from $n-1$ adversary piconets. 135
A.4 Packet error rate suffered by one piconet because of the interference from $n-1$ unsynchronized adversary piconets ($r = 366/625$) for different values of the traffic G. 135
A.5 Aggregated throughput. 136
A.6 Network topology with 9 interfering piconets. 137
A.7 Network topology with 49 interfering piconets. 138
A.8 Source process model. 138
A.9 Sink process model. 139
A.10 Jammer process model. 139
A.11 Simulation results for jammers transmitting with a power of 2 W. 140
A.12 Simulation results for jammers transmitting with a power of 20 mW. 140

B.1 Transceiver simulation model 144
B.2 Accumulated noise. 145
B.3 Accumulated noise simulation model. 145

List of Tables

3.1 Parameters used for the WiseNET SoC model. 30

5.1 Comparison for $L = 100$ s 80

6.1 Transceiver Parameters 113

7.1 Effective 32kHz quartz oscillation frequency 121
7.2 Measured static current consumption of XE88LC06A and XE1203 (3 V). 121
7.3 XE88LC06A and XE1203F two-chips solution. 122

A.1 n maximizing the aggregated throughput. 136

Chapter 1

Introduction

A Wireless Sensor Network (WSN) designates a system composed of numerous sensor nodes distributed over an area in order to collect information [88, 89, 3]. The sensor nodes communicate wirelessly with each other to self-organize into a multi-hop network, collaborate in the sensing activity and forward the acquired information towards one or more users of the information. Applications of sensor networks are numerous, ranging from environmental monitoring, home and building automation to industrial control [52].

The main requirements of sensor nodes is a low power consumption, allowing a lifetime of several years on a single small sized low-cost battery. As will be seen in the following chapters, a low power consumption is made possible, from the communication point of view, through a limitation of the traffic at application layer and through the acceptation of some latency when communicating. We will consider that the generated traffic will be low in average, not excluding however short high traffic periods.

The realization of wireless sensor networks presents many challenges. One of them is the design of communication protocols fulfilling their specific needs. A communication protocol stack is usually designed following the OSI 7 layer reference model [38]. The first four layers (1-4) are responsible of the information transport (single hop bit transfer at the physical layer, single hop packet transfer at the data link layer, multi-hop packet transfer at network layer, and reliable end-to-end packet transfer at transport layer). The remaining three layers (5-7) are responsible of the management of the communication. This dissertation deals with the design of energy efficient medium access control (MAC) for wireless sensor networks. A MAC protocol is located in the second layer (data link) of the OSI model.

This chapter introduces the different network topologies that can be taken by a sensor network and the requirements of sensor network applications. The main requirement, which is low power consumption, is discussed in more detail in the context of communication protocols. The field of this research is introduced, as well as its contributions.

1.1 Topologies

The classical topology that researchers consider when dealing with wireless sensor networks is the multi-hop topology illustrated in the top left graphics of Fig. 1.1. With this topology, the sensor network does not rely on any fixed infrastructure. The acquired information is forwarded towards a collection point called a *sink*. In this figure, a single collection point is depicted. Using multiple sinks in a sensor network is also possible. Applications of such a multi-hop topology

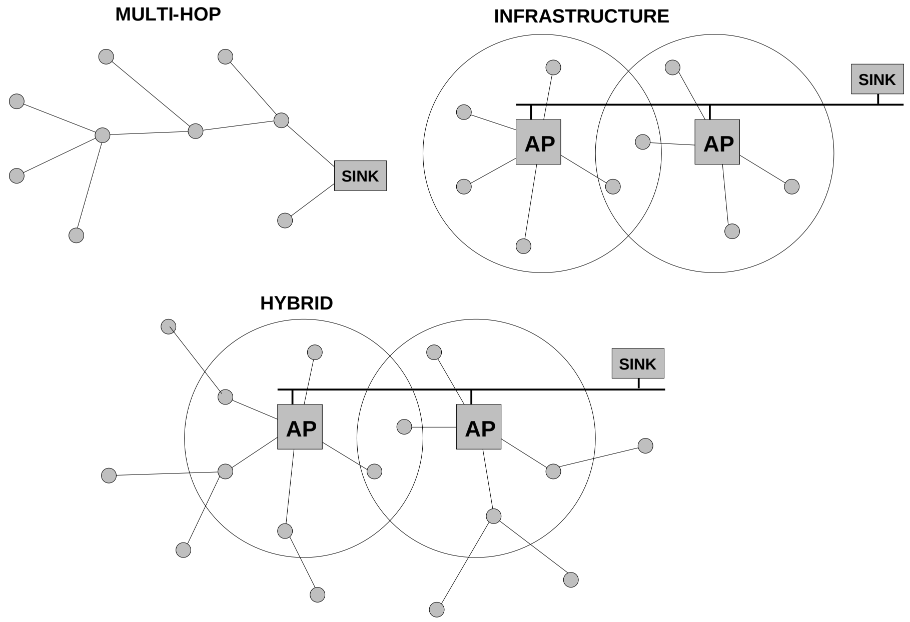

Figure 1.1: Sensor network topologies.

include for example soil monitoring in precision agriculture, structural health monitoring of civil infrastructure [129], volcanic activity monitoring [120] and wildlife observation [69]. When only one sink is collecting data from a sensor network, a traffic bottleneck may occur on the nodes near the sink. Data fusion [16] can be used with certain applications to mitigate such congestion. Another topology that can be of interest for sensor networks is the infrastructure topology illustrated in the top right graphics of Fig. 1.1. Each wireless sensor node must be located within range of a base station. Sensor nodes do not communicate together directly but through the base stations. Base stations are connected together and to the sink through a wired or wireless backbone network. The base stations and the backbone network form the so-called infrastructure. Base stations are usually assumed to be energy unconstrained. It will be seen later that this fact can be exploited to save energy on the sensor nodes. An infrastructure topology may for example be found in smart building applications, where a pre-existing Ethernet or Powerline network can be used as the backbone for the sensor network infrastructure. Finally, when mixing both topologies, one obtains the hybrid topology illustrated in the bottom of Fig. 1.1. The coverage of the base stations is extended through multi-hop communication. As each base station can collect part of the traffic, having more than one base station is an advantage from a capacity point of view.

Networks with a multi-hop topology are often called *ad-hoc* networks. The term ad-hoc refers to the self-organizing capability of networks, which is always a requirement of multi-hop networks. Self-organization is however a requirement that can be put on wired or wireless networks of any topology. For this reason, we will use the term multi-hop instead of ad-hoc when talking about topology.

1.2 Requirements

The requirements put on a sensor network vary from application to application. It is generally admitted [88, 89, 17, 3, 52] that the following requirements are set by most applications of sensor networks:

- **Low cost**. For some applications, a cost per node of 100 Euros is acceptable. For most applications, the cost should be reduced as much as possible into the direction of 1 Euro per node or less.
- **Long lifetime**. In many applications, it will not be possible or too expensive to replace or recharge the battery of sensor nodes. The required lifetime depends on the application. A lifetime of a few month can already be useful for certain measurement campaigns. In most applications, a lifetime of a few years or more is desired.
- **Large networks**. Many sensor networks can be expected to be composed only of tens of nodes. However, some applications may require sensor networks composed of hundreds or thousands of nodes. Sensor networks may also be large in the sense that the covered area is large.

The application requirements listed above lead to the following system requirements:

- **Low power consumption**: If sensor nodes are battery powered, the low cost requirement imply to use mass produced batteries of modest capacity. As a long lifetime must be reached with batteries of modest capacity, it is crucial to minimize the average power consumption of sensor nodes. The low cost and long lifetime application requirements translate into the most important system requirement, which is the low power consumption. An alternative explored today be researchers is to extract the energy out of the environment (e.g. indoor or outdoor light, vibration, acoustic noise) [89, 52]. Such techniques may provide an unlimited lifetime, but as the expected energy production is very small, the low power consumption of sensor nodes remains of the highest importance.
- **Low complexity of hardware and software**: Functions implemented in hardware should be as simple as possible because hardware complexity translates into larger chips, which are more costly and consume more energy. Software should be small and use a minimum amount of random-access memory to minimize the cost and power consumption of memory. Software should in addition minimize the power consumption of processing. Hardware-software co-design must be used to best allocate the required function into hardware or software. At the software level, a trade-off can sometimes be made between required memory and required processing for the implementation of the same function.
- **Multi-hop communication capability**. Because propagation loss is proportional at least to the square of the distance, it can be of interest from a power consumption point of view to forward a packet in a multi-hop fashion instead of transmitting it directly in a single hop. Another reason to use multi-hop communication is when the size of the deployed network exceed the maximum range of the transceiver. This maximum range is usually defined by the maximum power at which transmissions are allowed by regulation.
- **Self-organization**. Sensor networks should be able to self-organize into a network. This requirement is a consequence of the low cost requirement because self-organization mini-

mizes the deployment costs. In many applications (e.g. large multi-hop networks), manual configuration may even be intractable.

- **Scalability**. The communication protocols should be able to handle large networks.

1.3 Low power design

Low power design of wireless sensor networks must be addressed both at the level of hardware and software.

Low power hardware design can be tackled at the technological, logical and system levels [44]. The technological level refers to the integrated circuit design at lower voltage, lower frequency and higher integration. The logical level refers to power aware circuit design (e.g. clock gating). Low power hardware design at system level include the minimization of the energy consumed for inter-chip communication (e.g. through system-on-a-chip integration or memory caching).

The software running on a wireless sensor node is typically composed of an application and of a communication protocol stack.

To be energy efficient, an application should be aware of the power consumption impact of the services it request from the underlying hardware. Programmers should minimize the energy consumption while fulfilling the application requirements. Potential techniques include dynamic clock scaling [71] and power management. Dynamic clock scaling consists in adapting the processor frequency and voltage to the work load. Power management consists in turning off hardware peripherals that are not used. On mobile computers, the power management unit monitors the activity of the software and of the user to spin down the hard disk, turn off the display or enter a standby state. In a sensor network, the application shall ensure that peripherals such as sensors, actuators or memories are powered only when needed.

The power management of the communication interface is a task that needs to be tackled by the application and by all layers of the protocol stack. In certain systems, the application may know that no communication will be required for a long period of time. It may then decide to turn the transceiver off for that period. Turning off the communication interface when not used allows important gains because transceivers are often the highest power consumer of the node. However, there is still a need to communicate, and when traffic must be transferred, energy efficient mechanisms must be used by the communication protocol stack.

1.4 Energy efficient communication

This section briefly lists potential energy saving mechanisms at the individual layers. Further improvements can be achieved through cross-layer optimization [41].

1.4.1 Physical layer

A large effort has been devoted by the research community to improve the energy efficiency of wireless transmission between two nodes. Schurgers *et al.* have studied the tradeoff between power consumption and transmission delay when varying the modulation index [98]. They observed that with QAM modulation, the lowest energy consumption is reached with binary modulation. Holland *et al.* analyze in [46] the trade-off between bit rate and transmission range when the transmission power is constant. In a multi-hop communication environment, the selection of the transmission power impacts the transmission range that can be achieved and the amount of interference generated to others. The analysis of the optimal transmission range from an interference point of view has been studied by Tobagi and Kleinrock [110]. They observed that the transmission range which maximizes the expected progress results in a density of about 8 nodes within a circle of radius equal to the transmission range. In [13], Chen *et al.* show that there exist an optimal one-hop transmission power that minimizes the power cost of multi-hop transmission per meter. Another degree of freedom is the selection of error correction schemes. Redundancy can be added by the source to correct potential transmission errors (forward error correction, FEC). Several researchers have studied the trade-off in the choice of the error correction scheme as a function of the channel characteristics from an power consumption point of view [135, 75].

1.4.2 Data link layer

The data link layer is composed of two sublayers: the Medium Access Control sublayer (MAC) and the Logical Link Control (LLC) sublayer. A MAC protocol is an algorithm controlling the access of several nodes sharing a communication medium. The LLC layer is responsible for multiplexing upper layers and offering an optional communication reliability using error detection and Automatic Repeat Request schemes (ARQ). In wireless communication systems, ARQ is usually implemented in the MAC layer and the role of LLC is only protocol multiplexing. ARQ may be used in addition or as a replacement of FEC. Lettieri *et al.* study in [63] the trade-offs that may be made when combining FEC and ARQ. Ebert *et al.* analyze in [23] the trade-off between transmission power and required retransmissions when using ARQ.

Despite all the efforts invested in the design of low power communication circuits and in energy efficient data transmission schemes (e.g. modulation, channel coding, low power hardware implementation), the power consumption of a wireless communication transceiver remains today above 1 mW in receive mode, and much more in transmit mode. In order to achieve an average power consumption enabling years of lifetime on a low cost battery, it will be shown in chapter 3 that the average power consumption should be kept below 100 μW. It is hence mandatory, with today's technology, to turn the radio transceiver off most of the time. A transceiver may not listen to the channel all the time. A duty cycle of a few percent at maximum can be tolerated.

Because the transceiver of the sensor nodes may only be turned on during a small fraction of the time, there is a need for algorithms that organize the sensor nodes such that the source and the destination of a communication are turned-on at the same time. Because it is directly driving the radio transceiver, the MAC layer is ideally suited to address this task. Numerous techniques for power management at MAC layer exist. The issue of energy efficiency at MAC layer will be introduced in more details in chapter 2.

1.4.3 Network layers

At the network layer, routing protocols can be designed to distribute the traffic evenly among sensor nodes such that the average power consumption of every node is approximately equal. When the first nodes stop functioning, a multi-hop network may become partitioned and hence useless. Having an equal power consumption on every node can hence extend the overall network lifetime. Routing protocols may also exploit the high density of a sensor network to rotate the routing task among neighbors, letting the non-router nodes sleep (e.g. SPAN [11] and GAF

[130]). Another possibility of routing protocols is to manage the trade-off between using a low transmit power to reach a relay that is in the vicinity or a high transmit power to reach a relay that is located further away.

1.4.4 Transport, session, and presentation layers

Transport protocols can contribute to the energy efficiency of the system through the avoidance of congestion (which results in collisions and energy costly retransmissions). Another direction is to let the transport protocol shape the traffic into bursts allowing to power down the transceiver in-between bursts [7].

To save energy, both the session and presentation layers should minimize the introduced overhead. At the presentation layer, source coding can be used to compress the transmitted data and save energy through shorter transmissions.

1.5 Problem statement

The design of energy efficient physical layer communication and of energy efficient routing mechanisms have received a lot of attention in the research community [110, 63, 135, 23, 98, 46, 13, 75, 11, 130]. However, only few proposals [106, 89] had been made at the time this work was initiated for the medium access control protocol of wireless sensor networks. This dissertation deals with the design of energy efficient medium access control protocols fulfilling the specific needs of wireless sensor networks. A sensor network MAC protocol should be energy efficient. It must be simple to run on low cost processors with little amounts of memory. Schemes based on the use of an energy unconstrained base station should be avoided to permit multi-hop communication. It should contain a random access scheme to support self-organization. Algorithms should be local to allow scalability.

1.6 Contributions

The following contributions have been made with this dissertation:

The preamble sampling technique, previously proposed for paging systems, is a way to reduce power consumption when listening to an idle medium. A contribution of this dissertation was its analysis [26, 27] in a multiple access environment in combination with Aloha [1] and carrier sense multiple access (CSMA [58]). The classical renewal theory [58] used for the analysis of CSMA has been extended to provide estimates of the power consumption. The work on Aloha and CSMA with preamble sampling made within this thesis and published in [26] has lead the Berkeley research team developing TinyOS and the Mica motes to include the preamble sampling technique in their communication stack (low power listening in BMAC, see [109]).

Non-persistent CSMA with preamble sampling (NP-CSMA-PS) was shown to consume much more energy than time division multiple access (TDMA) when traffic is high. For this reason, at the beginning of the work, I initially proposed in [27] to use TDMA for the transport of frequent data traffic, and NP-CSMA-PS for the transport of the sporadic signalling traffic required for synchronizing sensor nodes into a TDMA schedule [27]. This proposal has been explored further experimentally by Reason and Rabaey [90].

The main contribution of this dissertation is the design and analysis of WiseMAC, a protocol that is building on CSMA with preamble sampling to achieve both a low power consumption in low traffic conditions and a high energy efficiency in high traffic conditions. Through piggybacking local synchronization information in every acknowledgement, WiseMAC is able to reach the high energy efficiency of a scheduled protocols such as TDMA without requiring the complexity and power consumption overhead associated with setting up a TDMA schedule. WiseMAC does not rely on a network wide synchronization and is therefore scalable. It was shown to perform several times better than state-of-the-art protocols proposed for wireless sensor networks either in terms or power consumption or in terms of latency. WiseMAC is able to transport sporadic and bursty traffic in addition to periodic traffic. This protocol has been developed, analyzed mathematically and through simulations, and validated through experimentation.

As part of the state-of-the-art survey, a classification of MAC protocols designed for wireless sensor networks has been proposed. This classification is novel in the sense that it captures the most important parameters differentiating WSN MAC protocols and permits to organize them into a tree structure.

Finally, during the analysis of existing protocols applicable for low power communication, the problem of interference between collocated slow-frequency hopping networks (such as Bluetooth [102]) has been studied (see Appendix A). A low bound on the packet error rate and a high bound on the aggregated throughput have been derived as a function of the number of collocated networks. This work [24, 25, 28] was the first to produce such analytical results for the Bluetooth system. Other researchers have since then extended and improved these results [72, 86, 66].

1.7 Thesis organization

Chapter 2 presents the state of research in the field of energy efficient medium access control for sensor networks. Models of a radio transceiver and a battery are introduced in Chapter 3. These models are used as a basis for the performance evaluation of MAC protocols. Chapter 4 analyzes the performance of TDMA and of CSMA with preamble technique, and shows why both protocols should be combined. Chapter 5 introduces WiseMAC, a protocol that presents the advantages of both TDMA and CSMA with preamble sampling. This chapter analyzes the performance of WiseMAC in a multi-hop network and chapter 6 analyzes the performance of WiseMAC for the downlink of an infrastructure network. Experimental results are presented in chapter 7 and chapter 8 gives conclusions.

Chapter 2

Energy Efficient Medium Access Control - State of the Art

This chapter presents a review of power saving techniques proposed for wireless communication systems by research and standardization at MAC layer, with an emphasis on the solutions designed specifically for wireless sensor networks.

2.1 Medium access control

The radio frequency spectrum is divided into frequency bands that are allocated to communication systems or groups of systems. A communication system can further channelize its frequency band using frequency division multiple access (FDMA), time division multiple access (TDMA) and spread spectrum techniques such as code division multiple access (CDMA) and frequency hopping. In addition, as the power of a transmission decay with the distance, it is possible to reuse the same resource at two locations given that they are sufficiently distant from one another (spatial reuse). Another possibility to obtain spatial reuse it to use directive antennas.

The allocation of the communication channels to transmitting devices can be fixed or dynamic. A fixed resource allocation is seen for example in radio broadcast systems. In a system where numerous computing devices are inter-connected through the wireless medium, it is impossible or at least very inefficient to allocate a channel to each device.

The role of a medium access control (MAC) protocol is to manage the dynamic allocation of one or several channels. MAC protocols may be classified according a number of characteristics [93, 43, 60]. The most fundamental characteristics are whether control is centralized or distributed and whether access in random, guaranteed or hybrid.

In a centralized MAC protocol, a central controller is in charge of managing the medium. Centralization simplifies the control algorithm but requires a star topology and usually puts more computing and power consumption demands on the central controller. Centralized MAC protocols are found in cellular systems (e.g. GSM [77]), wireless local area networks (e.g. IEEE 802.11 Power Save Mode and Point Coordination Function [79]) and personal area networks (e.g. Bluetooth [102] and IEEE 802.15.4 [82]). With distributed MAC protocols, the same algorithm is running on all nodes of the network. As they do not rely on the central control from a base station, distributed MAC protocols (e.g. CSMA [58], MACA [53], DBTMA [19]) are well suited for multi-hop networks. Because of their simplicity and efficiency, distributed MAC protocols are also of interest for WLANs (e.g. IEEE 802.11 Distributed Coordination Function

[79]). Some protocols designed for multi-hop networks use a clusterwise centralized control but rotate the role of central controller among neighboring nodes to balance the additional power consumption needed by the central controller among all nodes. It can be argued whether such protocols should be classified as distributed or centralized.

With a guaranteed access protocol (also called a contentionless or a conflict-free protocol) there can be no packet losses caused by collisions. Examples of purely contentionless MAC protocols include polling [102] and token passing protocols [80]. With a random access protocol (also called contention protocol), every transmission is subject to a probability of collision with other transmissions. The role of the contention protocol is to minimize the probability of collisions and to manage retransmissions. Examples of contention protocols include Aloha [1, 2] and CSMA [58, 73]. The combination of a random access protocol with a guaranteed access protocol is called a hybrid access protocol. The random access protocol can be used for the transport of delay tolerant data traffic and for the transport of resource allocation demands. Resource allocation is easily performed by a central controller, but distributed allocation is also feasible. The protocol used during the guaranteed access phase may for example be TDMA or polling. Examples of hybrid access protocols include PRMA [42] and DQRUMA [54].

A further characteristic of a MAC protocol is whether it can be operated with a single radio transceiver, or if an additional transceiver is needed (e.g. for the transmission of a busy tone to mitigate the hidden node effect [114] or for waking up the main transceiver [89, 99, 101]). As sensor nodes are very cost limited, needing more than one radio is a requirement that needs to be evaluated with care when designing a MAC protocol for wireless sensor networks.

2.2 Sources of energy waste at MAC layer

Before we proceed with the discussion of the techniques providing energy efficiency at MAC layer, it is of interest to have a look at the sources of energy waste that a MAC layer must address. Ye *et al.* [132] have identified the following four sources of energy waste:

- **Idle listening**: Idle listening refers to the active listening to an idle medium.
- **Overhearing**: Overhearing refers to the reception of data messages that are not destined to oneself.
- **Collisions**: Collisions occur when an interfering node transmits a packet in the vicinity of a node that is receiving another packet. Retransmissions will consume energy both on the transmitting and receiving sides.
- **Overhead**: The protocol overhead refers to the frame headers and the signalling protocol required to implement the medium access control protocol.

The power consumption of a transceiver when listening to an idle channel is the same or about the same as when receiving data. Many sensor network applications are foreseen to generate only little traffic. In can be expected that in many cases, the medium will remain idle for most of the time. In such cases, energy waste through idle listening can become very important.

The next most important source of energy waste after idle listening is overhearing, especially in dense networks. Singh *et al.* show in [103] the potential gains that can be achieved when mitigating only overhearing.

As traffic is expected to be low on average, collisions will be rare. However, bursty traffic periods can occur in many applications (e.g. event detection). Means to transport traffic bursts with a minimum of collisions must be designed with care to avoid congestion.

As every protocol, the MAC protocol of a sensor network must minimize its overhead. Such overhead includes the transmission and reception of periodic beacons, paging packets, wake-up preambles and acknowledgements.

2.3 Power saving schemes at MAC layer

In non-power saving systems, the word *access* in medium access control means *access for transmitting*. The MAC protocol must allocate the medium in an efficient and timely manner and prevent collisions between transmissions. The wireless nodes may listen to the channel all the time, except when they are transmitting. When power saving is used, the MAC protocol must also ensure that the destination of a transmission is awake during the transmission. The work access then means *access for transmitting and for receiving.*

There exist numerous methods to ensure that a node will be awake when it should receive data. Different solutions have been proposed depending on the system requirements. This section first briefly presents power saving schemes that have been proposed at MAC layer for WLANs, MANETs, WPANs and paging systems. As the requirements of these systems are different from those of sensor networks, the proposed protocols may not be reused as is. However, they may serve as sources of inspiration. The main proposals available in the literature for medium access control in sensor networks are introduced in more details.

2.3.1 Wireless Local Area Networks (WLANs)

Wireless local area networks (WLANs) are meant for the interconnection of computers. A WLAN is an infrastructure based network. MAC protocols for wireless local area networks are primarily designed to reach a high system throughput and to minimize the transmission delay. Low power consumption is left as a secondary requirement.

This section describes the power saving schemes in the IEEE 802.11 [79] and ETSI Hiperlan2 [56] standards. These protocols exploit the fact that the base station is energy unconstrained to save energy on the wireless nodes. Similar techniques are discussed in the following references: [104, 95, 94, 107]. In [12], Chen *et al.* compare the power consumption of WLAN MAC protocols in the high traffic regime. Krashinsky *et al.* propose in [59] a modification of the IEEE 802.11 power save mode that can reduce both the latency and the power consumption in the case of web access.

2.3.1.1 IEEE 802.11 infrastructure mode

IEEE 802.11 [79] can be operated either in infrastructure mode or in ad-hoc mode. The ad-hoc mode is meant for single cell connectivity. It will be discussed in more details when addressing MANETs. The basic medium access control used in IEEE 802.11, called the Distributed Coordination Function (DCF), is a variant of CSMA [121]. Using shorter inter-frame spaces than the mobile nodes, the base station can control the medium and provide a polling based guaranteed access. This optional protocol is called the Point Coordination Function (PCF). As of today, no implementation of the PCF function is available in off-the-shelf products.

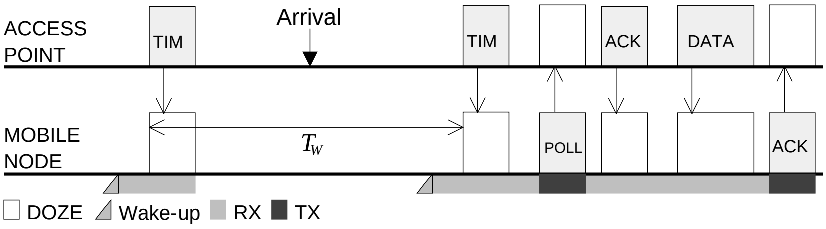

Figure 2.1: IEEE 802.11 infrastructure network, power saving mode for downlink communication.

In the downlink direction, the power saving scheme is based on the periodic transmission of beacons by the base station. This beacon contains the *traffic indication map* (TIM), which lists the wireless nodes for which data packets have been buffered. All wireless nodes in power saving mode have to wake up regularly to receive the beacon. If they discover their address in the TIM, they send a request to the access point (using the DCF contention protocol) to receive the buffered data following the procedure illustrated in Fig. 2.1. According to the standard, the request may be directly followed by the transmission of the data, but because the access point is usually unable to prepare the data for transmission within the specified delay (10 μs in DSSS 802.11b), the request is answered with an acknowledgement. Once it has received the acknowledgement, the wireless node stays in receive state and waits for the data transmission. After the successful reception of the data packet, it replies with an acknowledgement packet.

In the uplink direction, the procedure is trivial. As a base station is energy unconstrained, it may listen to the channel all the time. A node in power saving mode simply turns its transceiver on whenever it has a packet to send, transmits it using the DCF contention protocol, and goes back to sleep.

2.3.1.2 Hiperlan 2

The Hiperlan 2 standard [56] was released in year 2000. It was designed for the transport of both asynchronous and time critical traffic. It defines, as IEEE 802.11, two modes of operation. A centralized mode (using a base station) and a direct mode (for single cell ad-hoc connectivity). Hiperlan 2 is based on reservation TDMA. TDMA frames have a fixed duration of 2 ms. In a frame, slots are available for control, uplink, downlink, direct communication between two wireless nodes and random access. Resource allocation requests are transmitted in the random access period. The control field describes the resource allocation in the current frame and serves as a feedback on previous random access attempts.

An optional power saving mechanism is specified in [35]. Wireless nodes can request to enter the sleep state. Nodes in power save mode periodically wake up to listen to the control field. If the control field does not announce incoming traffic, the nodes goes back to sleep. The sleep period can be chosen to be 2^n times the frame duration, where $1 < n < 16$. The fact that larger sleep periods are divisible by the smaller ones allows to arrange all sleep groups to coincide periodically. This property can be exploited to transmit broadcast traffic to all power saving nodes at once. When a power saving node needs to initiate a transmission, it leaves the power saving mode.

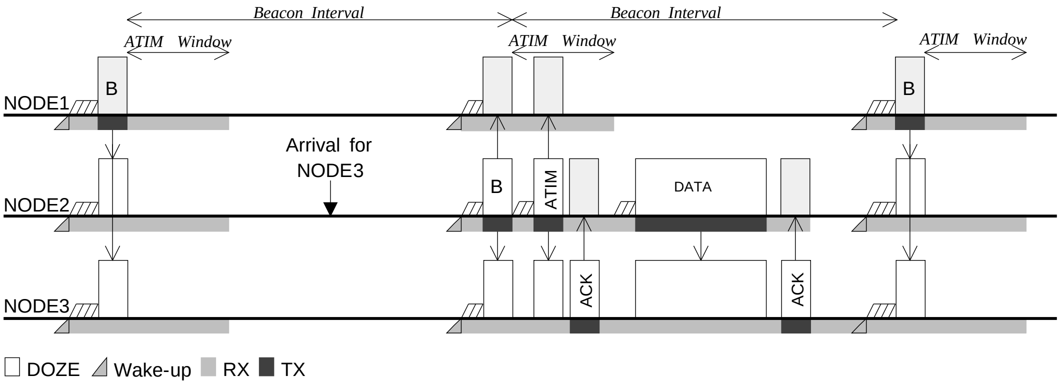

Figure 2.2: Power saving in an IEEE 802.11 ad-hoc network.

2.3.2 Mobile Ad-Hoc Networks (MANETs)

Mobile ad-hoc networks designate multi-hop networks of mobile computers. Much research has been devoted to ad-hoc routing within the MANET working group of the internet engineering task force (IETF). MANETs differ from sensor networks mainly in higher mobility, in higher throughput and in lower communication latency requirements.

2.3.2.1 IEEE 802.11 ad-hoc mode

The IEEE 802.11 standard specifies a mode of operation that does not rely on the use of a wired infrastructure. This ad-hoc mode was designed for the interconnection of computers located within range of each other (e.g. in a meeting room). Although it was not designed for multi-hop communication, most research on routing for MANETs assumes the use of this protocol at MAC layer.

In ad-hoc mode, mobile stations have to compete for the periodic generation of the beacon. This beacon can be exploited for discovery by new nodes entering the ad-hoc network. It is also useful to keep synchronized the nodes that are in power saving mode.

The power saving mode is based on the definition of a time window following the beacon during which all nodes have to be awake. Transmission towards nodes in power saving mode must be announced using ATIM packets (announcement traffic indication map) sent during the so-called *ATIM window.* Unicast ATIM packets are acknowledged. If a power saving node has received an ATIM packet announcing a broadcast packet or an unicast packet addressed to itself, it must remain awake for the full beacon interval to receive the data. Otherwise, it may go back to sleep. This procedure is illustrated in Fig. 2.2.

Several researchers have proposed enhancements to the IEEE 802.11 ad-hoc power-save mode. Woesner *et al.* study in [121] the trade-off between power consumption and the achievable throughput in the choice of the duration of the ATIM window. Variants designed specifically for multi-hop networks are presented in [115]. The scalability of the synchronization mechanism based on the distributed beacon generation in a multi-hop network is studied in [47].

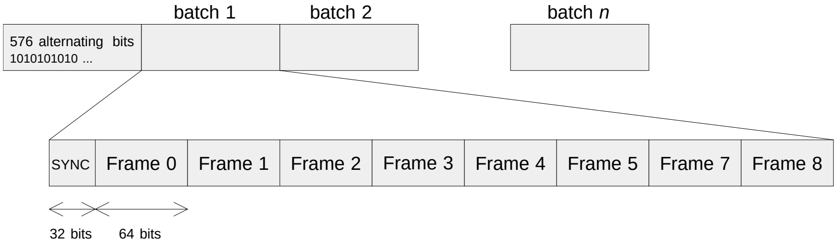

Figure 2.3: POCSAG paging system frame format.

2.3.2.2 Hiperlan 1

The Hiperlan 1 standard was published by ETSI in year 1996 [34]. Hiperlan 1 uses a distributed medium access control based on a variant of CSMA called EY-NPMA (Elimination Yield - Non-Preemptive Priority Multiple Access) [121]. Hiperlan was designed to provide ad-hoc multi-hop connectivity to mobile computers. It does not rely on a wired network infrastructure. Multi-hop communication is achieved via nodes that have taken the role of *forwarders*. The power saving scheme defined in Hiperlan 1 is based on a contract between two nodes: a *p-saver* and a *p-supporter*. The p-saver is active only during periodic intervals. The p-supporter must store packets destined to the p-saver and transmit them during the active intervals of the p-saver. In practice, as forwarders and p-savers will consume more energy that other nodes, it is likely that they will need to be powered from the mains.

Another interesting power saving scheme can be found in the framing. At high bit rate, an Hiperlan 1 transceiver needs to use a power hungry equalizer. Packets start at low bit rate with a 34 bits header that can be demodulated without equalizer. This header contains an 8 bits hash of the destination address. If a node has a matching hash, it starts the equalizer and receives the rest of the message. With this scheme, only 1/256 = 0.4% of nodes will overhear packets.

2.3.3 Paging systems

The goal of paging systems is to minimize the power consumption on the pagers and to maximizing the throughput. Latency is secondary.

2.3.3.1 POCSAG

POCSAG (Post Office Code Standardization Advisory Group) is a paging protocol that was developed by British Telecom in the seventies [70]. It was internationally adopted as the CCIR Radiopaging Code No. 1 [50]. The POCSAG frame format, illustrated in Fig. 2.3, is composed of a preamble of 576 alternating bits followed by a number of batches. A batch is composed of a 32 bits synchronization codeword followed by 8 frames of 64 bits. Pagers are assigned to one of 8 groups based on their address, and can be addressed only in the corresponding frame. The preamble is meant to wake up the pagers. In the absence of traffic, the pagers periodically wake-up and check for the presence of the preamble. When they detect the preamble, the pagers wait for the first synchronization word. Once synchronized, they may go back to sleep and wake-up periodically to listen to the frame corresponding to their address and to the (re-)synchronization codewords. This scheme permits a duty cycle below 7% in the absence of transmission, and below 15% during paging messages transmission [70].

2.3.3.2 FLEX and ERMES

In the beginning of the nineties, Motorola has introduced the FLEX protocol [70]. With the Motorola FLEX protocol, pagers remain continuously synchronized. A pager has to wake up every 30 seconds to listen to the frame corresponding to its address. The addresses of all nodes for which a message is scheduled are grouped at the beginning of the batches, allowing pagers for which no traffic is present to quickly go back to sleep.

In Europe, the European Radio Message System (ERMES) standard was specified by ETSI as a successor to POCSAG [33]. It uses a similar approach as FLEX (keeping the network synchronized and providing long sleep intervals). An improvement over FLEX found in ERMES consists in sorting the addresses. This allows the pagers to go back to sleep on average after half the duration of the address list.

2.3.4 Wireless Personal Area Networks (WPANs)

Wireless personal area networks (WPANs) designate short range networks centered around a person. WPANs differ from WLANs through the increased importance of the low power consumption and more modest requirements in terms of throughput. The two most important standards for WPANs are Bluetooth [102] and IEEE 802.15.4 [82]. Both are based on a star topology, where a central node is in charge of network coordination. Because network coordination requires additional energy, it can be expected that such networks will be centered around rechargeable devices such as mobile phones or portable computers.

2.3.4.1 Bluetooth

Bluetooth is a digital wireless data transmission standard in the 2.4 GHz ISM band aimed at providing a short range wireless link between laptops, cellular phones and other devices [102]. The air interface modulation is Gaussian FSK with a raw bit rate of 1 Mb/s (3 Mb/s in the next version). The communication topology between Bluetooth nodes is point-to-multipoint, where a master communicates in time division duplex with several slaves using the polling protocol, forming a so-called piconet. In order to tolerate interference which can readily arise in the 2.4 GHz band, a slow frequency hopping scheme is used, where all nodes of a piconet hop together among 79 frequencies at each packet slot.

Bluetooth defines three power saving modes: hold, sniff and park. In the hold mode, the node leaves the piconet for an agreed period of time, possibly to save power, but also to discover or connect to other nodes. In sniff mode, a node periodically wakes up to receive potential traffic. In park mode, a node leaves the piconet but remains synchronized with the hopping sequence such that it may join the piconet rapidly once necessary. It must periodically wake up to refresh the synchronization. The piconet master, being in control of everything, may simply sleep whenever it has nothing to do.

The use of Bluetooth for multi-hop networking is possible, as a node may simultaneously be part of multiple piconets. This however places severe constraints in the operation of the respective piconets. Several research groups have experiments multi-hop networking based on Bluetooth [55, 62]. The main drawback of Bluetooth for ad-hoc networks is the long time that is needed to discover neighbors and connect to them. This high latency is caused by the use of the frequency hopping technique. Bluetooth specifies that a device that wants to be discoverable listens during 10 ms on a fixed frequency every 1.28 seconds. A different channel, chosen from 32

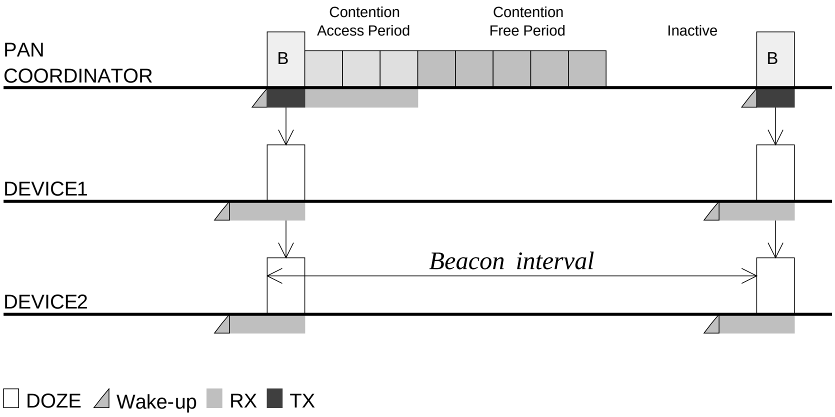

Figure 2.4: IEEE 802.15.4 beacon-enabled superframe structure.

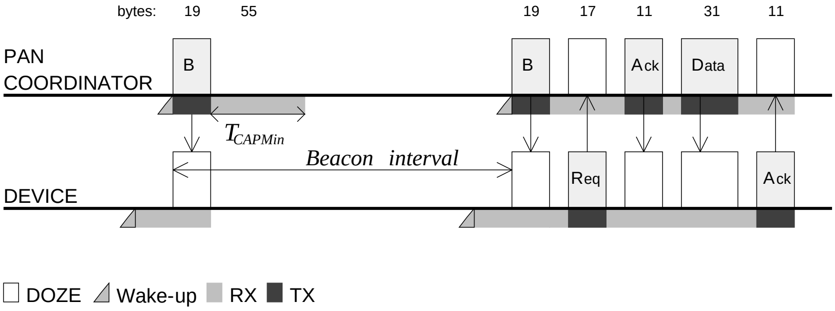

Figure 2.5: Downlink data transfer in a beacon-enabled IEEE 802.15.4 network.

channels, is used every time. A discovering device sends packets during 10.24 seconds hopping over these 32 channels. At worse, it takes about 10 seconds to reach every node. The discovery delay (inquiry delay in Bluetooth terminology) was reported to be of 2.2 seconds on average [55]. The scheme chosen for Bluetooth puts little overhead on the discoverable devices (they need to listen for 10 ms over 1.28 s, i.e. less than 1 % of the time) at cost of a large number of transmissions by the discovering device and a high latency in the discovery procedure.

After the discovery of a neighboring node, another procedure is required to setup a connection. This procedure, called paging, requires again the transmission of a burst of packets over the 32 paging channels and results in an additional delay.

2.3.4.2 IEEE 802.15.4

An IEEE 802.15.4 [82] network has a star topology. The central node is called the personal area network (PAN) coordinator. The other nodes are called devices. A PAN can be beacon-enabled or non beacon-enabled. The MAC protocol used in a non beacon-enabled network is unslotted CSMA/CA. The PAN coordinator must be always listening to the channel to receive uplink data from devices and requests for the download of potential downlink data[1]. As a battery operated device may not listen all the time to the channel, non beacon-enabled operation is not

---

[1] Direct data transfer between two devices controlled by a coordinator is allowed in the standard (chapter 5.4.2.3) but the measures to achieve the synchronization between the two devices needed for low power peer-to-peer data transfer are left unstandardized.

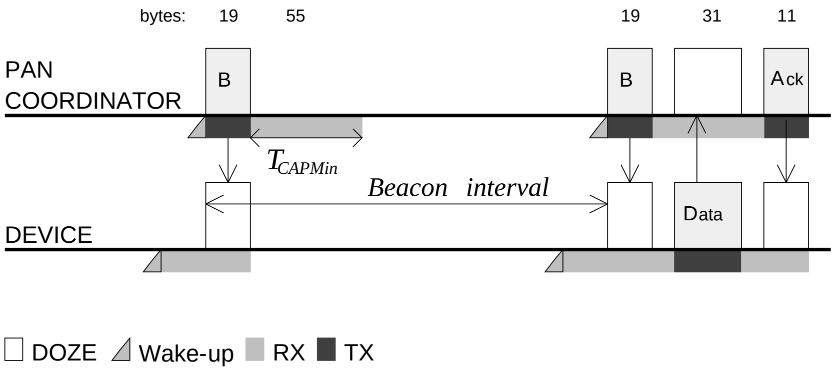

Figure 2.6: Uplink data transfer in a beacon-enabled IEEE 802.15.4 network.

suited for multi-hop networks. We will hence consider further only beacon-enabled PANs. In a beacon-enabled PAN, the coordinator regularly transmits a beacon. The MAC protocol of a beacon-enabled PAN is composed of a Contention Access Period (CAP) and of a Contention Free Period (CFP) as illustrated in Fig. 2.4. When bandwidth reservation is desired in the CFP, a device sends a GTS (Guaranteed Time Slot) reservation request to the PAN coordinator during the CAP using slotted CSMA/CA.

Assuming that there is no collision between multiple devices, a transmission from the PAN coordinator to a device in power saving mode follows the procedure illustrated in Fig. 2.5. This procedure is identical to the one used in the power save mode of IEEE 802.11 infrastructure networks: The coordinator buffers incoming packets destined to devices in power saving mode. The address of devices for which a packet has arrived is inserted in the following beacon. Every device in power saving mode regularly wake up to receive the beacon. If they find their address, they send a data request to the coordinator. The data request is acknowledged by the coordinator. It then transmits the data to the device, which responds with an acknowledgement.

Assuming that there is no collision between multiple devices, a transmission from a device to the coordinator in a beacon-enabled network follows the procedure illustrated in Fig. 2.6. The coordinator must listen during at least 440 symbols after having sent the beacon to receive potential data requests or uplink data packets. 440 symbols correspond to 440 bits (or 55 bytes) with BPSK modulation and to 1760 bits (or 220 bytes) when using O-QPSK. The size of beacon, data request and acknowledgement packets is respectively of at least 19, 17 and 11 bytes.

Multi-hop communication is possible with IEEE 802.15.4. The ZigBee Alliance is specifying a routing layer that shall be used above the IEEE 802.15.4 MAC layer [4]. If IEEE 802.15.4 is used for multi-hop communication, a relay node will typically act as a device for its parent and as a coordinator for its children. It will hence have to periodically transmit a beacon to its children and receive a beacon from its parent. Such a relay node will have to periodically listen to the 19 bytes of the beacon sent by its parent, and periodically transmit a beacon of 19 bytes, followed by a listen period of 55 bytes after the beacon transmission. This adds up to a periodic transmission of 19 bytes and a periodic listening during 74 bytes. The overhead caused by the periodic reception and transmission of beacons is very large, especially if the power consumption in transmit state is important. MAC protocols designed specifically for multi-hop wireless sensor networks, as will be presented in the next section and in the rest of this dissertation, can achieve a much lower overhead. In addition, when used for energy constrained applications, IEEE

802.15.4 introduces a topology limitation. Indeed, the coordinator may have a maximum of 7 devices in power save mode due to the size limitation of the traffic indication map address field in the beacons.

2.3.5 Wireless sensor networks

The design of MAC protocols targeted specifically for wireless sensor networks has recently gained a lot of attention from the research community. Numerous proposals have been made in the past few years. The proposed protocols all have in common that their main goal is to save energy by allowing the sensor node to sleep most of the time. They differ in how to organize the sensor nodes wake time. Fig. 2.7 shows a classification of MAC protocols proposed for wireless sensor networks, including the proposals resulting form this dissertation (NP-CSMA-PS and WiseMAC). This classification is an original contribution of this thesis. Classifications presented in [93, 43] address MAC protocols in general and do not highlight the most important parameters of MAC protocol designed for WSN. Langendoen *et al.* give in [60] a table classification of WSN MAC protocols based on three parameters: the number of channels used (single, double, multiple), the type of organization (random, slots, frames) and the type of notification (listening, wakeup and schedule). The classification presented in Fig. 2.7 uses a tree structure to give a synthetic view of the protocol classification according to the most important parameters. The first classification parameter is whether the protocol is scheduled or unscheduled.

With scheduled MAC protocols, all sensor nodes in the network are synchronized. Communication takes places during predefined time slots and sensor nodes sleep in between their activity slots. Time slots are allocated either to links between pairs of sensor nodes, to individual sensor nodes or to groups of sensor nodes. When time slots are allocated to links, collisions between packets cannot happen. Access is guaranteed. When a time slot is allocated to a node, this slot can be reserved for transmission by this node, for reception by this node or allow both traffic directions. When a time slot is allocated to a node for reception, one must find means to control access between several nodes that may want to communicate to the same node during the same time slot. They may simply transmit their data packet using the Aloha or CSMA contention protocols, in which case we talk of random access. Instead of transmitting the data packet in contention, they may also transmit a resource allocation request in contention. The data is later transmitted using a guaranteed access protocol. In such a case, we talk of hybrid access.

Unscheduled MAC protocols are based on the use of a wake-up scheme to ensure that the destination of a packet is awake when the packet is transmitted. Communication in an unscheduled multi-hop network must be based on random access. The proposed protocols can be classified depending on the mechanism used as the basis of their wake-up scheme. A node can periodically listen to see if another node wants to talk to him, periodically transmit to see if another node wants to reply to him, or use some means that are external to the radio channel used for data communication.

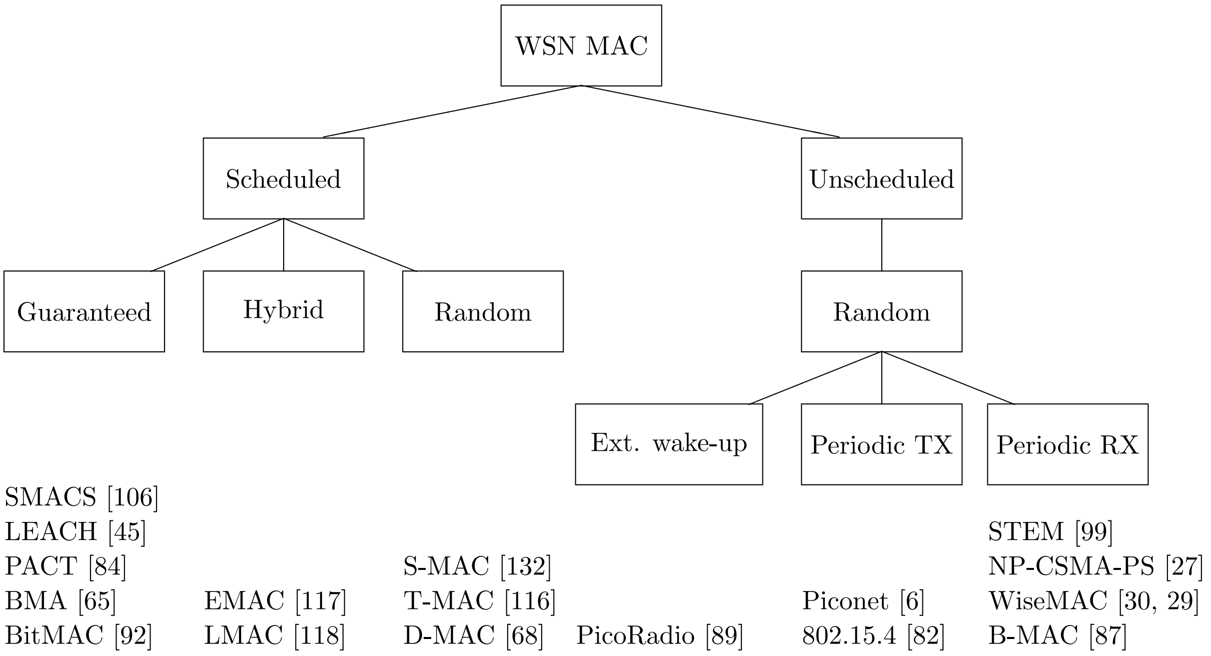

Figure 2.7: A classification of MAC protocols for wireless sensor networks.

2.3.5.1 Scheduled MAC Protocols

Guaranteed Access

Sohrabi *et al.* have proposed SMACS, a TDMA based solution [106]. SMACS defines a procedure for the discovery of neighbors and the allocation of a frequency and a time slot for communicating with them. To simplify the complex problem of network synchronization and TDMA schedule setup, they proposed to allocate the time slots between nodes at different frequencies, randomly chosen from a large pool.

LEACH (Low Energy Adaptive Clustering Hierarchy) is a clustered based protocol [45]. Clusterheads are assumed to be able to increase their transmission power to reach a base station in one hop. They organize the communication with their neighbors in a TDMA schedule, collect measurement data and forward the aggregated information to the base station. To balance the power consumption among all nodes, the clusterhead role is rotated. Because with LEACH every node must be able to communicate with a base station, this protocol does not fulfill the multi-hop communication requirement in the cases where multi-hop communication is desired to extend the transmission range of base stations.

Pei *et al.* have proposed the Power Aware Clustered TDMA protocol (PACT) [84]. As in LEACH, clusterheads are assumed to be able to transmit gathered data directly to the base station. PACT is power aware in the sense that the cluster head role is rotated taking into account the remaining energy of nodes. The TDMA scheduling algorithm proposed in PACT is based on the exchange of information two-hops away. TDMA frames are composed of control slots and data slots. Every node must listen to all control slots. Each node has an allocated control slot and uses it to announce data traffic. The weaknesses of this protocol are the fact that all nodes must be able to communicate directly with the base station (as in LEACH) and that nodes must listen during all control slots in every TDMA frame, which represent a large overhead.

Li *et al.* have proposed to use a Bit-Map Assisted protocol (BMA) for intra-cluster communication [65]. At the beginning of each TDMA frame, each node having a packet to send transmits during the allocated bit. After the bit-map section, the clusterhead broadcasts a transmission schedule. Ringwald *et al.* have designed and experimented a bit-map based protocol called BitMAC [92]. A discussion of the basic bit-map scheme can be found in [111]. The application of BMA for sensor networks is of little interest in low traffic conditions as collisions are rare even with a simple protocol like CSMA. BMA can be of interest during traffic peaks, but the high synchronization accuracy required to operate BMA is likely to translate into a high synchronization overhead and a high associated power consumption overhead.

Hybrid Access

EMAC (EYES MAC) is a clustered demand-assignment TDMA protocol proposed by Hoesel *et al.* in [117]. Each clusterhead owns a timeslot. A slot is subdivided into request, control and data sections. The request section is used by the members of the cluster to demand the allocation of the slot. The control section is used to announce downlink traffic. Cluster members have to periodically listen to the control section. Cluster heads have to periodically listen to the request section and periodically transmit the control information. The control information includes the time slot occupancies as perceived by a clusterhead in its neighborhood. Having received the slots occupancy information from all its neighboring cluster heads and their neighbors, a node can choose a free time slot.

In [118], Hoesel *et al.* describe LMAC (Lightweight MAC), a variant of EMAC. With LMAC, clustering is not used anymore. Time slots are subdivided into a control and a data section. Every node owns a time slot and periodically transmits control information indicating the identity of the slot owner, the number of hops to the gateway and, if a data packet is following, the destination address and the length of the data packet. As clustering is not used, the request section present in EMAC slots has been removed. Every node has to listen to the control information section of every time slot. If a node sees its address or the broadcast address, it listens to the data section. The number of time slots in a TDMA frame limits the maximum connectivity in the network. As every node must listen to the control section of every time slot, the overhead of LMAC is large.

Random Access

Ye *et al.* have proposed S-MAC (Sensor-MAC) [132]. As illustrated in Fig. 2.8, this protocol defines sleep intervals in which all the nodes of the network sleep, and listen intervals in which traffic announcement signalling can occur. Because listen intervals are relatively large, only a loose synchronization is required among neighboring nodes. The listen interval contains three sections. The first section is reserved for synchronization messages. The second and third sections are dedicated to the exchange of request-to-send and clear-to-send packets. These packets are used to reserve the medium and signal to the destination to remain awake for the data transmission. With S-MAC, one must select the frame duration (i.e. the total of the listen and sleep intervals), as a trade-off between the average power consumption and the transmission delay. S-MAC exploits the concept of fragmentation to transmit large messages in an energy

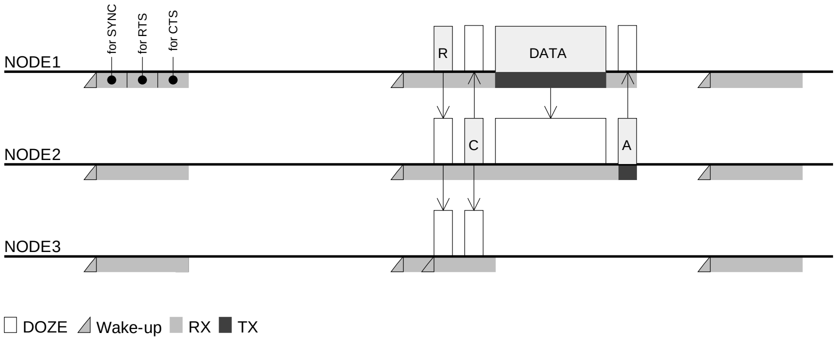

Figure 2.8: S-MAC.

efficient way.

Van Dam *et al.* have proposed T-MAC (Timeout-MAC) [116]. T-MAC is an improvement of S-MAC. In the T-MAC protocol, the length of the active period is dynamically adapted to the volume of traffic, using a timeout. The active period is ended whenever physical and virtual carrier sensing find the channel idle for the duration of the time-out. A similar idea was independently proposed by Ye *et al.* in [133].

D-MAC is a variant of S-MAC proposed by Lu *et al.* in [68]. D-MAC reduces the transmission delay compared to S-MAC in applications where data is gathered towards a sink from sensors arranged into a tree structure. Delay is reduced by cascading the listen intervals of the sensor nodes.

2.3.5.2 Unscheduled MAC Protocols

External Wake-up

Rabaey *et al.* have chosen in the PicoRadio system a hardware based solution to wake up a destination node [89]. They suggest the use of a separate super-low-power wake-up radio that will switch the main radio on at the start of the data packet. This solution is of great interest as it would provide small hop latencies. The wake-up preamble being short, this method would also preserve channel capacity. The development of such a super-low-power wake-up radio consuming only a few tens of μW being still a challenge, solutions using conventional radio transceivers remain of interest. If such a wake-up radio becomes available, it may also be envisaged to use it in combination with low power MAC protocols, to reduce even further the base power consumption to a few μW.

Another external wake-up technology can be found in the field of RFIDs. Passive RFIDs consume absolutely no energy in sleep mode. They are woken up and powered through an incoming inductive wave [14]. This technique cannot unfortunately be applied to multi-hop sensor networks because of the high energetic cost of the wake-up signal at the transmit side.

Periodic Transmission

Bennett *et al.* have proposed in Piconet [6, 40] a wake-up scheme based on the periodic broadcast of one's identity. This scheme, illustrated in Fig. 2.9, demands to all nodes to regularly wake-up, broadcast their identity, listen for a short while, and go back to sleep if no

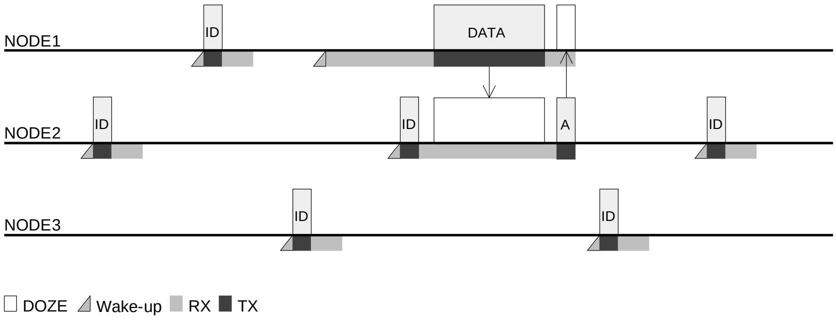

Figure 2.9: Piconet.

data reception has started. This approach can provide very low power consumption when little traffic is present and when the wake-up interval is large. Data packets are sent when the intended destination is listening. At that time, having a very low duty cycle, the other neighboring nodes are very likely to sleep. Hence, this protocol provides a solution to the problem of overhearing. Limitations of this protocol appear when a wake-up interval in the order of 100 ms is desired in order to have a short hop transmission delay. The first problem is then that the background traffic caused by the broadcast of the ID packets becomes non-negligible and can cause frequent collisions with data packets. Secondly, the average power needed to transmit the ID packet and to listen for potential replies becomes important. Note finally that the periodic ID broadcast in Piconet is similar to the periodic beacon transmission in IEEE 802.15.4.

Periodic Listening

Schurgers *et al.* have proposed STEM [99]. This protocol uses two channels: one paging channel and one traffic channel. Most of the time, the network is expected to be in the monitoring state, and only the paging channel is used. In case of an alarm, for example, a path on the data channel is opened throughout the network, where communication occurs using a regular wireless protocol (not low power). In STEM, the paging channel is implemented at the receiver side by regularly listening to the channel during the time needed to receive a paging packet. A transmitter that wants to page one of its neighbor will repeat a paging packet containing the destination address, until a reply is received. The operation of the paging channel is illustrated in Fig. 2.10. This protocol provides a low power consumption in the absence of traffic, the paging channel consuming little energy. The weakness of this protocol is mainly its inefficiency to transport small amount of periodic or sporadic traffic. Note that the wake-up scheme proposed by STEM can be seen as a simplified version, at a single frequency, of the discovery procedure used in Bluetooth.

NP-CSMA-PS [26, 27] and WiseMAC [30, 29] are proposals resulting from this dissertation. They will be discussed in the following chapters.

B-MAC is a CSMA based protocol. It has been developed at Berkeley and implemented on TinyOS [109]. To reduce the idle listening power consumption, B-MAC uses the preamble sampling technique described in [26]. In the context of B-MAC, preamble sampling is called Low Power Listening (LPL). The particularity of B-MAC is to have been implemented on TinyOS with a rich configuration interface, allowing the application programmer to configure the protocol mechanisms (clear channel assessment, acknowledgements, backoffs and LPL).

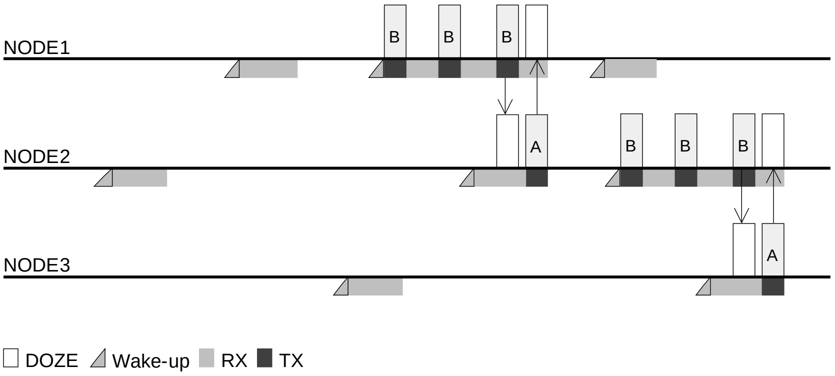

Figure 2.10: Operation of the paging channel in STEM.

2.4 Summary

Excluding schemes using an external wake-up hardware, power saving schemes proposed at MAC layer are either based on a periodic transmission and/or on a periodic reception. The network nodes may be synchronized or not.

When a central node is available (wireless LANs, paging systems, or clustered sensor networks), a powerful scheme consists in periodically broadcasting a traffic indication map. All nodes must be synchronized to wake up for receiving the traffic indication map. The same scheme can also be distributed to be used for mobile ad-hoc networks of computers (IEEE 802.11 ad-hoc power save protocol) and sensor networks (S-MAC). Synchronization becomes more challenging in such cases. The S-MAC protocol can be seen as an extreme case of the IEEE 802.11 ad-hoc power save protocol, an optimization for low power consumption in low traffic conditions. The S-MAC listen interval can be compared to the IEEE 802.11 ATIM window. The main differences between them are first that S-MAC exchanges only sporadically SYNC messages, while IEEE 802.11 requires a beacon transmission at the beginning of each interval, and second that the listen interval in S-MAC is chosen to allow for only one data transmission, as opposed to several with IEEE 802.11.

Several protocols proposed for sensor networks provide a guaranteed access through the use of TDMA. TDMA appears at first glance as a very appealing protocol for wireless sensor networks. It causes neither overhearing nor collisions and sensor nodes may sleep in between assigned communication slots. However, it suffers from two drawbacks. First, it is only efficient when transporting periodic traffic, however many sensor network applications will generate sporadic or bursty traffic. Secondly, the setup of the TDMA schedule between nodes can be a complex issue requiring a complex protocol implementing some distributed consensus. Such a protocol can be expected to generate numerous signalling packets, which consumes energy. To avoid this complexity, some proposals put constraints elsewhere. For example, through the use of another frequency for each link [106] or by limiting the network density to fit within a chosen number of time slots [118]. Pre-configuration would be an alternative but goes against self-organization and mobility.

Among the unscheduled protocols, Piconet is an example of protocols based on a periodic transmission. A node having traffic to send will listen until its receives a signal. STEM is the opposite example. It is based on a periodic reception. A node having traffic to send transmits a burst of messages until it receives a reply. As a transceiver usually consume more in transmit state than in receive state, periodically receiving (STEM) will consume less energy than periodically transmitting (Piconet). However, transmitting a packet will consume less energy with Piconet, as listening for a while consumes less than transmitting a wake-up burst. In sensor networks, as traffic can be expected to be low on average, it is better to put the higher cost on the transmit side, and use a scheme based on periodic reception.

This dissertation will study protocols based on preamble sampling, the technique used in the POCSAG paging system. Preamble sampling is similar to STEM in its spirit. It does however consume less energy.

Beside academic research results, one can already observe the availability of several industrial solutions for wireless sensor networks. MicroStrain Inc. uses a direct communication link between sensor nodes and a base station in the Wireless Web Sensor Networks (WWSN) [74]. MillennialNet proposes the iBean multi-hop network solution, with which gateways are connected to an unconstrained source of energy and used to relay information towards the end points [91]. A solution with energy unconstrained gateways is also used by EnOcean [31]. As of today, it appears that no industrial solution is being proposed that can provide ultra-low power multi-hop communication. Multi-hop communication using intermediate energy unconstrained gateways is today a mature technology. Multi-hop communication with battery powered relays is the challenge.

Among the MAC protocols proposed for wireless sensor networks at the time the WiseMAC protocol was proposed, the most relevant ones were S-MAC and T-MAC. Other MAC protocols proposed at that time included SMACS, LEACH, Piconet, IEEE 802.15.4 and STEM. SMACS and LEACH are (TDMA based) scheduled guaranteed access protocols. The comparison with these protocol is not convenient as WiseMAC is a unscheduled random access protocol. A lower bound on the power consumption of TDMA based protocol will be used as a benchmark through the definition of an ideal protocol. Piconet and IEEE 802.15.4 are based on a periodic transmission. Their overhead is much larger than the one of S-MAC. STEM uses a wake-up scheme that is similar to preamble sampling. Because STEM is based on a periodic listening phase of the size of a packet, its overhead is the same as the one of S-MAC. STEM however suffers from the drawbacks, as compared to S-MAC, of requiring the transmission of a paging packet burst to wake-up the destination, and to use a non-power-saving MAC protocol for the transport of data packets. These arguments lead to the choice of S-MAC and its variant T-MAC as benchmarks when studying WiseMAC.

Among the protocols proposed recently, the EMAC and LMAC protocols are clearly associated with an overhead larger than the one of S-MAC. D-MAC is another variant of S-MAC. The power consumption of D-MAC is identical to the one of S-MAC, while the transmission delay is reduced for the special case of applications were data is gathered towards a sink from sensors arranged into a tree structure.

Chapter 3

Battery and Transceiver Models

3.1 Introduction

To evaluate the performance of an ultra-low power medium access control protocol, one needs a very precise model of the radio transceiver power consumption. Existing models available in the literature (e.g. [37]) did not describe the energy consumed when switching between stable states. When the average power consumption is in the order of a few tens of microwatts, it is required to take into account the energy spent to turn the radio on and to switch between receive and transmit states. Those energy amounts are not negligible anymore.

To translate an ultra-low average power consumption into a lifetime, it is required to use a battery model that takes into account the self-discharge of the battery. A simple battery model has been designed to evaluate the expected lifetime assuming a constant self-discharge.

Section 3.2 presents the battery model and section 3.3 the radio transceiver model.

3.2 Battery model

There exist many different energy sources that can be used for sensor network applications. Energy can be taken from a primary battery (non-rechargeable), from a secondary battery (rechargeable), or taken out of the environment (e.g. light and vibrations energy scavenging [89, 52]). Secondary batteries may be used in combination with energy scavenging to store the excess energy production. In applications where energy scavenging is not possible or not sufficiently reliable, one must rely on primary batteries.

The two primary battery technologies that are best suited for low power wireless communication applications, in terms of cost, shelf-life and delivered peak current, are alkaline and lithium. Lithium/manganese batteries provide a constant voltage of 3 V over their lifetime [22], while alkaline batteries have a sloping discharge curve starting at 1.6 V and ending at about 1 V [21]. The discharge curve for both technologies is illustrated in Fig. 3.1.

When designing battery powered systems with a target lifetime counted in years, it is important to take into consideration the self-discharge. Alkaline batteries present a self-discharge per year of several percents of their capacity, while lithium batteries loose about one percent or less [57]. For example, a Duracell alkaline battery will loose 5% of its capacity during the first year and 2% during the subsequent years [21]. An Energizer E91 AA alkaline battery will loose 20% of its capacity in seven years, giving a discharge rate of 3% per year [36]. In contrast, a Duracell lithium/manganese battery will loose only 3% of its capacity after 5 years, giving

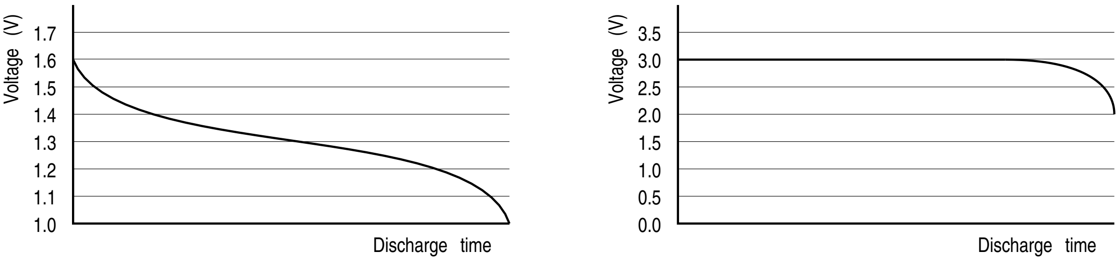

Figure 3.1: Typical discharge curve for alkaline/manganese batteries (left) and for lithium/manganese batteries (right).

a self-discharge of 0.6% per year [22]. Although lithium/manganese batteries have superior performances (stable voltage and smaller leakage), we will consider the use of alkaline batteries because of their lower price. We will consider throughout this dissertation that the energy source is a single AA alkaline battery with $C = 2600$ mAh initial capacity. Let us assume that the sensor node is operating between 1 and 1.6 V. At the start of the battery life, the excess voltage could theoretically be exploited to reduce the consumed current. We will consider it as lost. The energy capacity of the battery becomes $E = C \cdot 1.0 = 2.6$ Wh. We assume a constant leakage equal to 10% of the initial capacity during the first year. This assumption is quite conservative. It has been made because we wanted to take into account low cost batteries. This corresponds to $P_{Leak} = 0.1 \cdot E/(24 \cdot 365) = 27\ \mu$W. For simplicity, the leakage is assumed to remain constant over the following years. With an average power consumption P, the battery will be empty at time T as given by

$$\begin{aligned} T(P) = \frac{E}{P + P_{Leak}} &= \frac{E}{P + 0.1 \cdot E/(24 \cdot 365)} \text{ hours} \\ &= \frac{E}{24 \cdot 365 \cdot P + 0.1 \cdot E} \text{ years} \end{aligned} \tag{3.1}$$

This model implies a maximum lifetime of 10 years even without load (i.e. for $P = 0$). Fig. 3.2 shows the node lifetime as a function of the average consumed power. It can be observed that the average power consumption must be below 100 μW to reach a lifetime of more than 2 years.

3.3 Radio transceiver model

3.3.1 Model parameters

3.3.1.1 Power consumption and transition delays

When designing energy efficient communication protocols, it is very important to precisely model the static and dynamic power consumption of the used radio transceiver. One must identify the different states that the transceiver can be in, the power consumed in those states as well as the transition delays and the average power consumption during transitions.

Feeney *et al.* have presented in [37] an energy consumption model for IEEE 802.11 PC-cards, where the transceiver is assumed to be in one of four states: *sleep*, *idle*, *receive* and *transmit.*

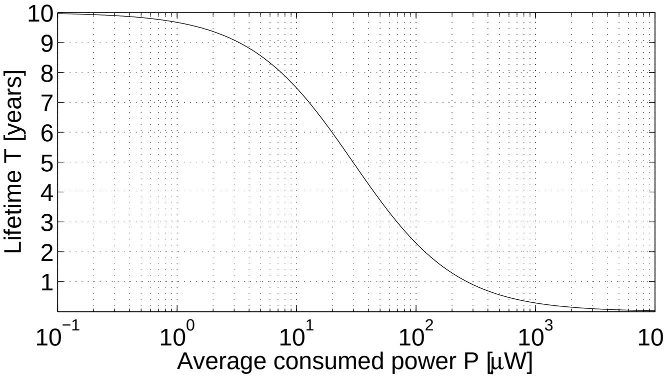

Figure 3.2: Lifetime of a sensor node using a single 2.6 Ah alkaline battery as a function of the average consumed power.

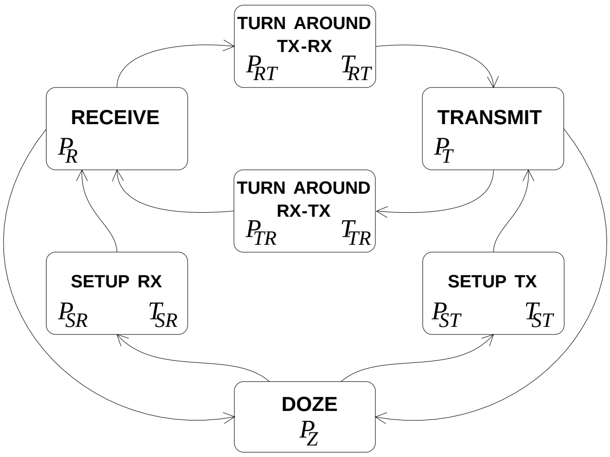

Figure 3.3: Transceiver states, power consumption and transition delays.

The current consumption of a Lucent WaveLAN PC Card (11 Mbps IEEE 802.11 DSSS) was measured as 10, 156, 190 and 284 mA in the sleep, idle, receive and transmit states, respectively. The sleep state is a power saving state in which the transceiver cannot receive. The sleep state differs from the state in which the transceiver is completely turned off by the shorter time needed to reach the idle and transmit states. In the idle state, the transceiver is ready to receive data. When the start of a data packet is detected, the receive state is entered. The current consumed in the receive state is slightly higher than in the idle state, because of the additional processing required for data reception. The transition delays between the idle, receive and transmit states are very small with an IEEE 802.11 DSSS card. They have hence not been addressed in the presented model. This model is not applicable for the analysis of low duty cycling protocols because it does not consider the delay and power consumption of transitions.

The transceiver model that will be used in this dissertation is illustrated in Fig. 3.3. It includes the following states:

DOZE The transceiver is not able to transmit nor receive, but is ready to quickly power-on into the receive or the transmit state,

RECEIVE The transceiver is listening to the channel, receiving data or trying to demodulate data out of a noisy or idle channel,

TRANSMIT The transceiver is transmitting data,

SETUP RX The transceiver is setting up into the receive state,

SETUP TX The transceiver is setting up into the transmit state,

TURNAROUND TX-RX The transceiver is turning around into the receive state,

TURNAROUND RX-TX The transceiver is turning around into the transmit state.

Compared to the model presented in [37], transition states have been added to consider the amount of time and energy spent in switching between stable states. It is important to consider the setup transitions, because the energy spent in the setup phase may dominate the energy consumption in very low traffic conditions. It is important to consider the turn-around phase, because the turn-around delay in the receive to transmit direction affects the collision probability in carrier sensing protocols. The idle state considered in [37] has been eliminated because transceivers suitable for wireless sensor network applications have a low baseband complexity and do not display a different power consumption when listening to the channel and when receiving data. The term *doze* has been used in our model to designate the state called *sleep* in [37]. No assumption is made in this model on what is actually running in the doze state. The classical choice would be to have the radio electronics, including the radio quartz, turned off in doze state. Another option would be to keep the radio quartz running while in the doze state, allowing a very quick startup at the cost of a higher power consumption.

T_{SR} and T_{ST} denote the setup time required to turn on the transceiver into respectively the receive and transmit states, starting from the doze state. P_{SR} and P_{ST} denotes the average power consumed during the corresponding setup phases. T_{TR} denotes the turn-around time from transmit to receive and T_{RT} denotes the turn-around time from receive to transmit. P_{TR} and P_{RT} denotes the average power consumed during the corresponding turn-around phase.

To simplify analytical expressions, we define $\widehat{P}_X = P_X - P_Z$ as the increment in power consumption caused by being in the state X, as compared to the doze state. With this definition, it will be possible to compute the average power consumption as the sum of P_Z and the proportion of time spent in the other states multiplied by the incremental power consumption in those states ($\widehat{P}_R$, $\widehat{P}_T$, $\widehat{P}_{SR}$, $\widehat{P}_{ST}$, $\widehat{P}_{TR}$, $\widehat{P}_{RT}$).

3.3.1.2 Other parameters

To calculate the maximum transmission range that can be achieved, one must know the transmission power and the receiver sensitivity. The transmission power $\mathcal{P}_T$ is defined as the power at the chip output. The sensitivity of a transceiver is the minimum power level $\mathcal{P}_S$ at the chip input, with which a signal can be demodulated at a given bit rate and bit error rate. Losses in the antenna and between the antenna and the chip must be taken into account separately in the link budget. These parameters will be used in chapter 5, when discussing communication in a multi-hop environment.

When dealing with protocols using the received signal strength indicator (RSSI), one must define the required integration time T_I needed to obtain an accurate measurement of the RSSI.

The shortest required duration to obtain a precise measurement is the duration of a symbol. A larger integration time may be necessary depending on the transceiver capabilities and the desired level of noise rejection.

We finally denote with R the bit rate of the transceiver.

3.3.2 WiseNET SoC model

CSEM has developed a system-on-chip targeted for wireless sensor network (WiseNET) applications [85, 32]. The WiseNET SoC includes a sensor interface, the CoolRISC 8 bits microcontroller and an FSK radio transceiver. It has been designed to operate from 1.6 to 1 V, allowing the use of a single alkaline battery as the energy source.

In receive mode the WiseNET SoC consumes $P_R = 2.1$ mW (868 MHz channel, 1 V power supply) and provides a sensitivity of $\mathcal{P}_S = -108$ dBm (25 kb/s, 10^{-3} BER) [85]. The current consumption in transmit mode and the effective transmitted power depends on the supply voltage. For the 868 MHz channel, the current consumption varies between 30 and 40 mA and the transmitted power varies between 6 and 11 dBm when varying the supply voltage between 1 and 1.5 V. Assuming that the voltage provided by the alkaline battery decreases linearly, we will consider the average values of $\mathcal{P}_T = 8.5$ dBm for the transmitted power and $P_T = 35$ mW for the consumed power. $P_T = 35$ mW results from multiplying the average current with 1 V, as the battery model assumes the loss of the energy provided by a supply voltage above the minimum, although it is not lost when transmitting. With a battery model that does not assume this loss, the average power consumption when transmitting would be 44 mW.

The CoolRISC microcontroller executes one instruction per cycle. At the maximum operation frequency of 6.4 MHz, it can provide 6.4 MIPS. When active, the microcontroller consumes 60 μW per MIPS. A dedicated radio interface hardware (similar to the BitJockey in [124]) provides a byte interface between the radio and the microcontroller, relieving the microcontroller from computing intensive operations such as bit and frame synchronization. The core is woken up only for writing or reading the next byte. The power consumption of the controller will hence be neglected.

The power consumption in doze state is in the order of $P_Z = 5$ μW at 1 V. This includes the circuit leakage current as well as the current consumed by a 32 kHz oscillator. An optimized setup procedure to switch from the doze state to the receive or transmit states consists in turning on later the blocks that turn on more rapidly. Fig. 3.4 illustrates the setup procedure into receive state, and shows the current consumption during the different phases. The first phase consists in turning on the high frequency quartz oscillator (XTAL). It needs at maximum 1.5 ms to settle. Most of times, it settles within 1 ms. The current consumption of the WiseNET SoC amounts to 300 μA and 60 μA respectively during and after the settling phase. At time 1.5 ms, the voltage controlled oscillator (VCO) and the receiver baseband (RX BB) are turned on. The current consumption climbs to 1.4 mA. After 100 μs, the radio frequency receive block (RX RF) is turned on. The current consumption reaches 2.1 mA, the nominal value in receive state. An additional waiting time of 100 μs is required to be able to receive data or start a measurement of the Received Signal Strength (RSSI). The total time needed to setup into receive state amounts hence to $T_{SR} = 1.7$ ms. The setup energy integrates to 0.7 μJ, which gives an average power consumption of $P_{SR} = 0.4$ mW during T_{SR}. The setup procedure into transmit state is similar. We have $P_{ST} = 0.4$ mW and $T_{ST} = 1.7$ ms. The turn-around time between receive state and transmit state amounts to $T_{RT} = 100$ μs. Assuming that the receiver baseband is kept running

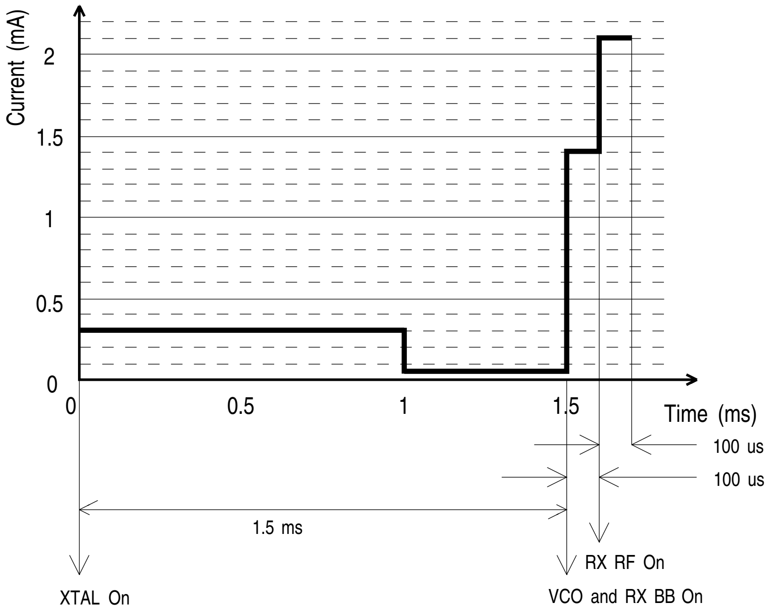

Figure 3.4: Current consumption of the WiseNET SoC during setup phase into receive state.

Table 3.1: Parameters used for the WiseNET SoC model.

$P_Z = 5\ \mu$W	$P_{SR} = P_S = 0.4$ mW	$T_{SR} = T_S = 1.7$ ms	$\mathcal{P}_T = 8.5$ dBm
$P_R = 2.1$ mW	$P_{ST} = P_S = 0.4$ mW	$T_{ST} = T_S = 1.7$ ms	$\mathcal{P}_S = -108$ dBm (BER=10^{-3})
$P_T = 35$ mW	$P_{TR} = P_R = 2.1$ mW	$T_{TR} = T_T = 0.1$ ms	$R = 25$ kbps
$U = 1.0$ V	$P_{RT} = P_R = 2.1$ mW	$T_{RT} = T_T = 0.1$ ms	$T_I = 100\ \mu$s

during transmit, the same value can be used for the transmit to receive turn-around time T_{TR}. During the turn-around phases, the power consumption is assumed to be equal to the power consumption is receive state ($P_{RT} = P_{TR} = P_R$). It is assumed to operate the transceiver at $R = 25$ kbps, giving a bit duration of 40 μs. To reach a reasonable accuracy when measuring the received signal strength, the integration time should be $T_I = 100\ \mu$s. As radio and controller are integrated into a single chip, the time to read the RSSI value can be neglected. The parameters used for the WiseNET SoC model are summarized in table 3.1.

As the setup time and power consumption into receive and transmit states is identical for the WiseNET SoC, we will simplify the notation and use T_S and P_S instead of T_{SR}, T_{ST} and P_{SR}, P_{ST} respectively. As the turn-around time is equal in both directions, we will use the notation T_T instead of T_{TR} and T_{RT}.

3.4 Conclusion

In this chapter, an alkaline battery model has been defined, assuming a constant self-discharge equivalent to loosing 10% of the initial capacity in one year. This model can be considered as quite conservative. Recent performance figures indicate that a lower self-discharge can be achieved with newer alkaline batteries.

A model of the WiseNET SoC radio transceiver has been defined. This model includes the time and energy spent during the transitions between stable states. The energy consumption of an optimized power-on procedure has been considered.

Both models will be used for the evaluation of MAC protocols in the following chapters.

Chapter 4

Spatial TDMA and Non-Persistent CSMA with Preamble Sampling

4.1 Introduction

MAC protocols designed for wireless sensor networks can be classified into scheduled and unscheduled protocols (see chapter 2). This chapter analyzes the performance of a scheduled protocol (Spatial TDMA) and of an unscheduled protocol (NP-CSMA-PS). The analysis of their respective strengths and weaknesses will show how they could complement each other and, when combined, provide an energy efficient solution.

As was seen in chapter 2, many scheduled protocols proposed for wireless sensor networks use the time division multiple access (TDMA) protocol and allocate communication time slots to either links, nodes or group of nodes. In a multi-hop network, the same slot can be reused at a two sufficiently distant locations. In such a case, one speaks of spatial TDMA [78]. We will consider, as in [106], a spatial TDMA network where time slots are allocated to individual links. Such a network is illustrated in Fig. 4.1. As an example of spatial reuse, the pairs of nodes on the upper left corner and on the lower right corner share the same time slot.

Section 4.2 analyzes the power consumption of sensor nodes in an established S-TDMA network. The initial synchronization of the network nodes and the allocation of time slots to the individual links requires the exchange of signaling messages. This signalling protocol requires some transport mechanism that does not rely on the synchronization of the network nodes. Such a transport mechanism is the NP-CSMA-PS protocol, which is a combination of non-persistent carrier sensing multiple access (NP-CSMA) with the preamble sampling technique (PS). This protocol will be analyzed in section 4.3. The combination of both will finally be discussed in section 4.4.

4.2 Spatial TDMA

4.2.1 Scenario

Let consider that the task assigned to the sensor network is to periodically acquire sensor measurements and forward them to a sink. The amount of traffic flowing through the nodes in the network depends on the number of active sources in their subtrees. In most cases, one can expect the average traffic to increase when approaching the sink. The concept of data fusion

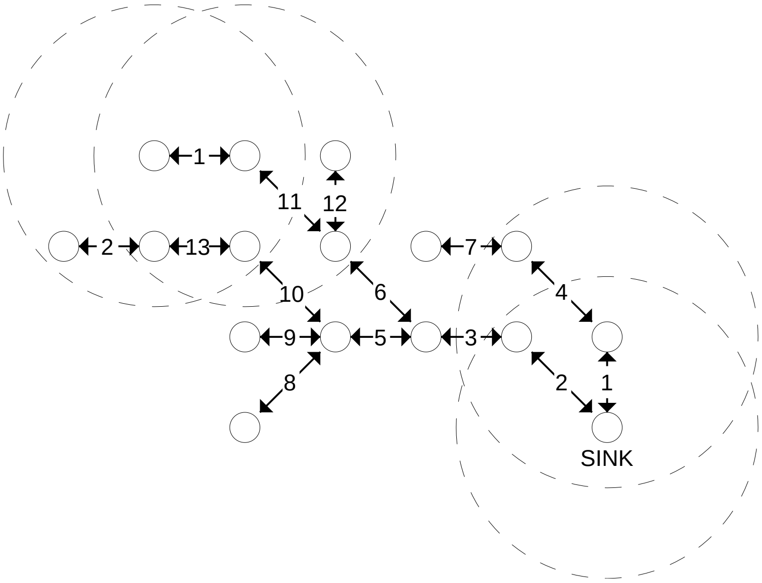

Figure 4.1: Spatial TDMA.

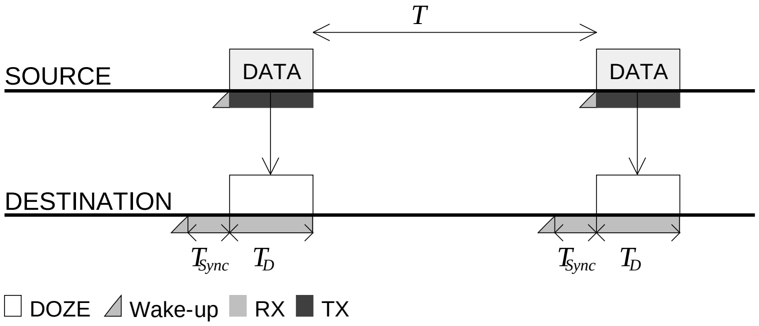

Figure 4.2: TDMA communication with earlier listening for clock drift compensation.

[16] permits to mitigate this effect and provide a similar traffic throughout the network. Even when using data fusion, network topology will have an impact on the traffic of individual nodes. Indeed, nodes having several children will receive more packets than they send. Such aspects are linked to the design of the application and of the routing layer. They will not be addressed as we are interested in the MAC layer. We will focus on the analysis of the basic forwarding problem. A node receives and sends exactly one packet within a period of L second.

4.2.2 Required synchronization interval

Let us assume that a destination node synchronizes itself with a source node whenever it receives a message. Assume that a message was exchanged at time zero, and that a message needs to be transmitted at time T (see Fig. 4.2).

Because the clocks at the source and the destination may drift away due to quartz imprecision, the destination must start listening earlier than the agreed time. The quartz imprecision, denoted with θ, is given as a tolerance in ppm (parts per million). Typical tolerance values for low cost crystals remain below 100 ppm. For example, the frequency tolerance of the Saronix

NTF3238 is specified to be within ± 20 ppm [96]. The worst case duration of the required synchronization period T_{Sync} can be computed using the model shown in Fig. 4.3. Because the clock of the source node can be imprecise, the effective time at which the transmission will start can vary between an early bound T_{S1} and a late bound T_{S2}. These bounds can be computed in the following way: In order to wake up at the target time T, the source's micro-controller counts $n = T/T_{Quartz} = Tf_{Quartz}$ cycles, where f_{Quartz} is the theoretical clock frequency of the source and T_{Quartz} the theoretical clock period. Because its real clock frequency can vary between $f_{Quartz}(1-\theta)$ and $f_{Quartz}(1+\theta)$, the lower bound is $T_{S1} = \frac{n}{f_{Quartz}(1+\theta)} = \frac{Tf_{Quartz}}{f_{Quartz}(1+\theta)} = \frac{T}{(1+\theta)}$. Similarly, we get $T_{S2} = \frac{T}{(1-\theta)}$.

In order to detect the start of a message, the destination must listen between T_{S1} and $T_{S2} + T_{Prop}$. The propagation time is included in the upper bound to consider nodes that are as far from one another as the transceiver range permits. For the lower bound, the worst case situation is found with the source and the destination located close to each other. The propagation time would be zero in that case. Because the clock of the destination is imprecise, one must program guard times in order to be certain to meet the deadlines. To be certain to wake-up not later than T_{S1}, the destination must target a wake-up at time T_D. The clock drift at the sender and receiver are assumed to be independent. This assumption is pessimistic, as the quartz inaccuracy is related to the device temperature and aging, two parameters that are likely to be correlated between the nodes in a network. The four extreme cases are illustrated at the bottom of Fig. 4.3. The worse case is found when the destination is waking up early and the source is sending late.

The duration of the interval $[T_{S1}; T]$ is $T - T_{S1} = T - \frac{T}{1+\theta} = \frac{\theta T}{1+\theta} = \theta T(1 - \theta + \theta^2 - ...) \approx \theta T$ for $\theta \ll 1$. Similarly, the duration of the intervals $[T_{D1}; T_D]$, $[T_D; T_{D2}]$ and $[T; T_{S2}]$ is approximately equal to θT. The required synchronization period in the worst case is hence of duration $4\theta T + T_{Prop}$. For a radio system with a maximum range of 100 meters, we have $T_{Prop} = 100/3 \cdot 10^8 = 0.33$ μs at maximum. The propagation time will be negligible if $4\theta T \gg T_{Prop}$, i.e. if the interval between communications T is much larger than $\frac{T_{Prop}}{4\theta}$. With a very accurate quartz with $\theta = 10$ ppm, we have $\frac{T_{Prop}}{4\theta} = \frac{3.3E-7}{40E-6} = 8.25$ ms. In sensor network applications, the minimum period between message transmissions can be expected to be much larger than 8.25 ms. The propagation time can hence safely be neglected. In the worse case, the duration of the synchronization period is hence simply

$$T_{Sync} = 4\theta T \tag{4.1}$$

Assuming that the imprecision of both quartzes is uniformly distributed over $[-\theta; +\theta]$ and taking the average over all nodes, one would obtain an average duration for the synchronization period of $2\theta T$. However, because a communication takes place between two given nodes, it is safer to consider the worst case duration $T_{Sync} = 4\theta T$ and not its average value. Note finally that a source may learn the relative drift rate between its own clock and the clock of its destination. The synchronization accuracy can thereby be improved.

4.2.3 Power consumption

Assuming the forwarding of one packet of duration T_D every L seconds, the power consumption of TDMA can be expressed as

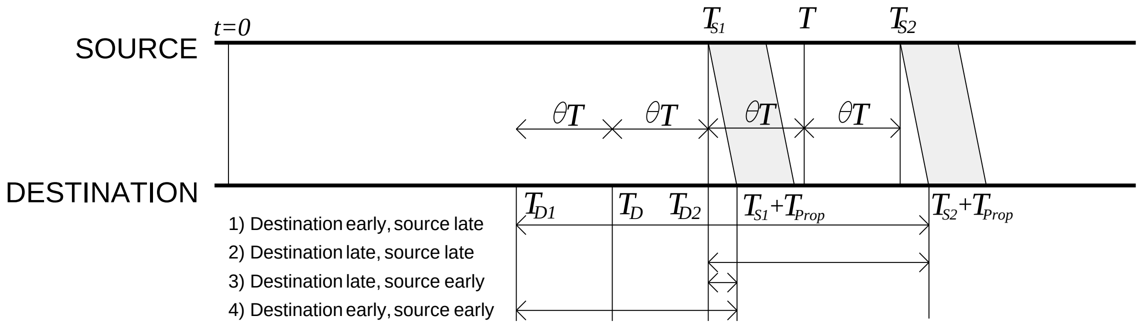

Figure 4.3: Required synchronization period due to source and destination clock drifts. T is the target transmission time. T_{S1} and T_{S2} are the early and late limits for the start of the transmission by the source. T_D is the target time for listening at the destination. T_{D1} and T_{D2} are the early and late limits for the effective start of the listening phase by the destination.

$$P_{TDMA} = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R(T_{Sync} + T_D + T_T) + \widehat{P}_T T_C}{L} + \frac{\widehat{P}_S T_S + \widehat{P}_T T_D + \widehat{P}_R(T_T + T_C)}{L} \tag{4.2}$$

In this expression, P_Z is the power consumption in doze state. $\widehat{P}_S$, $\widehat{P}_R$, $\widehat{P}_T$ represent respectively the power consumption increment compared to the doze state caused by being in the doze to receive transition state, in the receive state and in the transmit state (see the transceiver model in section 3.3). The second and third terms in expression (4.2) represent the average power increment needed respectively to receive a packet and re-transmit a packet. In the second term, $\widehat{P}_S T_S$ is the energy to setup the transceiver into receive state. $\widehat{P}_R(T_{Sync} + T_D + T_T)$ is the energy required to listen to the channel until the message starts, to receive the message, and to turn-around from receive to transmit. $\widehat{P}_T T_C$ is finally the energy required to transmit an acknowledgement message of duration T_C (where the letter C means control). The third term is derived similarly. We have the energy to setup into transmit state $\widehat{P}_S T_S$, the energy to transmit the data message $\widehat{P}_T T_D$ and the energy to turn-around into receive state and to receive the acknowledgement $\widehat{P}_R(T_T + T_C)$.

Using $T_{Sync} = 4\theta L$ (expression (4.1)), we have

$$P_{TDMA} = P_Z + \frac{2\widehat{P}_S T_S + \widehat{P}_R(T_D + T_C + 2T_T) + \widehat{P}_T(T_D + T_C)}{L} + 4\theta\widehat{P}_R \tag{4.3}$$

$2\widehat{P}_S T_S + \widehat{P}_R(T_D + T_C + 2T_T) + \widehat{P}_T(T_D + T_C)$ represents the minimum energy consumption required to forward a packet. The overhead required to keep the synchronization is $4\theta\widehat{P}_R$. Very interestingly, this overhead is not dependent on the volume of traffic. The power consumption of TDMA, using the radio parameters of the WiseNET transceiver, is shown in the upper part of Fig. 4.4 as a function of the forwarding interval. The lower part shows the translation of the consumed power into a lifetime, using a single AA alkaline battery modelled as presented in section 3.2. For the quartz, a tolerance of $\theta = 30$ ppm has been considered. With a power consumption of 100 μW (the upper bound of our power budget), the TDMA protocol is able to forward packets at a rate of up to 1 packet every 6 seconds. It is interesting to note that

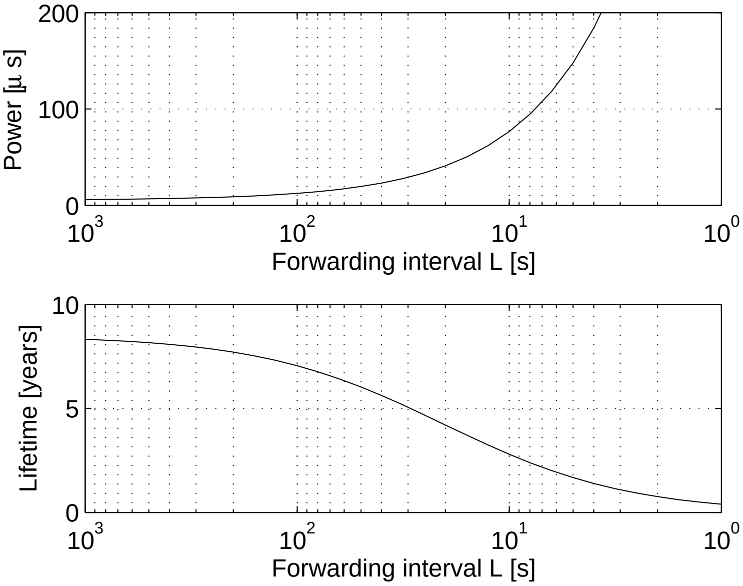

Figure 4.4: Power consumption and lifetime with the TDMA protocol, when forwarding 60 bytes packet every L seconds using the WiseNET transceiver.

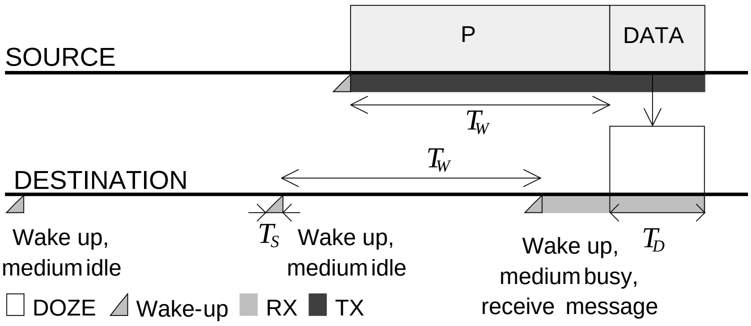

Figure 4.5: Preamble Sampling.

the cost of keeping the synchronization is only of $4\theta\widehat{P}_R = 0.25\ \mu$W. This cost is only relatively significant when the traffic is very low. Compared to the leakage current of an alkaline battery, it is negligible.

4.3 Non-persistent CSMA with preamble sampling

4.3.1 Preamble sampling

The preamble sampling technique, illustrated in Fig. 4.5, consists in periodically sampling the medium to check for activity. In this context, *sampling* means measuring the received signal strength. The sampling period is denoted with T_W, where the letter W refers to *wake-up*. The time during which the received signal strength is measured is denoted with T_I. If a node finds the medium busy, it continues to listen until it receives a data packet or until the medium becomes idle again. At the transmitter, a wake-up preamble of size equal to the sampling period T_W is transmitted ahead of every data packet to ensure that the receiver will be awake when data transmission begins.

The preamble sampling technique is used for example in the POCSAG paging system (see section 2.3.3.1). The WiseNET SoC has been designed to permit an efficient implementation of preamble sampling [61].

Using the transceiver model presented in section 3.3, one can compute the energy increment required to sample the medium as the energy to power-on the transceiver $\widehat{P}_S T_S$ and the energy to sense the channel $\widehat{P}_R T_I$. In the absence of traffic, the power consumed by the sampling activity is given by

$$P = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} \tag{4.4}$$

With a transceiver optimized for fast RSSI measurements, such as the WiseNET transceiver, the time needed to sense the channel T_I is only of a few radio symbols. The strength of the preamble sampling technique lies in the fact that T_I is short. As was seen in chapter 2, other wake-up mechanisms using a conventional transceiver (i.e. without external wake-up hardware [89]) are based on periodically listening to the channel for the duration of a few bytes (e.g. in EMAC, LMAC, S-MAC, T-MAC, D-MAC, IEEE 802.15.4, STEM). Some schemes require in addition the periodic transmission of a few bytes (e.g. IEEE 802.15.4, Piconet).

Because preamble sampling requires only a measurement during a few symbols, this wake-up scheme provides, in the absence of traffic and using a conventional transceiver, the lowest possible power consumption. However, this low idle listening power consumption is paid for by the wake-up overhead when traffic is present. The wake-up preamble causes a power consumption overhead at the source, at the destination, and not to forget, at all overhearing nodes.

4.3.2 Carrier sensing protocols

Carrier sensing protocols attempt to avoid collisions through the observation of the medium before to transmit. On wired networks, it is in addition possible to listen while transmitting and thereby detect collisions (see Ethernet [73] and IEEE 802.3 [81]). With a wireless transceiver, because of the large power difference between the transmitted signal and the potentially received interfering signal, collisions are difficult to detect. In practice, carrier sensing is only a collision avoidance scheme in wireless systems. Collisions (and transmission errors caused by noise) are detected using an acknowledgement scheme.

Sensing the channel is a way to reduce the probability of collisions, but not to avoid them. Collisions remain possible because of the *hidden node effect* and the remaining *vulnerability period.*

The hidden node effect refers to the fact that a node A wanting to send data to a node B might be unable to sense the transmission from a node C, while node B will be disturbed by node C. Carrier sensing is performed at the transmitter location, while interferences matter at the receiver's location. The hidden node effect will be discussed when considering multi-hop networks in chapter 5. Here, to simplify analytical computations, we will consider that every node is in range of all other nodes.

The second potential source of collisions with carrier sensing protocols stems from the fact that signals take time to propagate, that a transceiver needs time to switch from the receive to transmit states and that it needs time to measure the signal strength on the channel. Let us

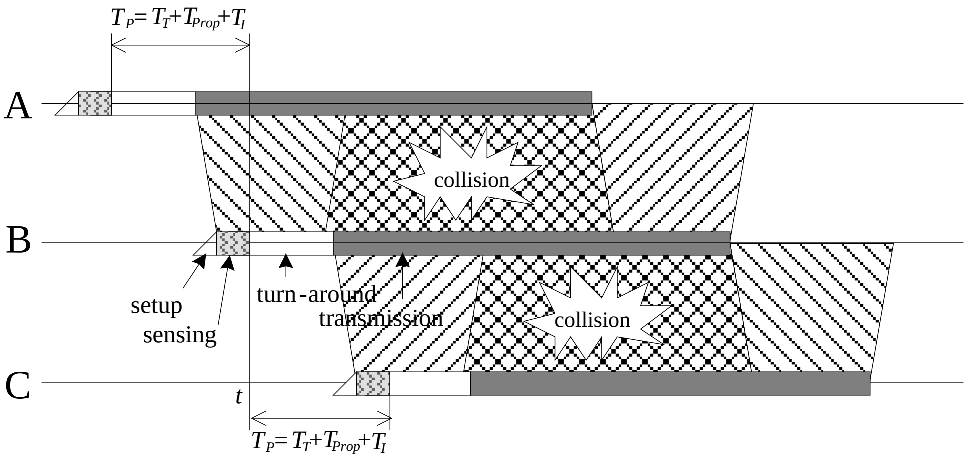

Figure 4.6: Residual vulnerability period with CSMA.

consider the time diagram shown in Fig. 4.6. Let T_{Prop} be the propagation time. As earlier defined in section 3.3, T_T is the time to turn-around the transceiver between receive and transmit state and T_I is the time needed to sense the channel.

If a node B senses the medium idle at time t, it will switch the transceiver from receive mode into transmit mode and then start the transmission. If a node A has done the same up to $T_T + T_{Prop} + T_I$ before, its transmission will not have been sensed by B. Similarly, a node C might sense the channel idle and decide to transmit its message up to $T_T + T_{Prop} + T_I$ later. The total duration of the vulnerability period is hence $2(T_T + T_{Prop} + T_I)$. In [58], only the propagation time was taken into account. Turn-around and sensing were assumed to be instantaneous. Sensor networks are short range systems. For a transmission range of 100 meters, the propagation time amounts to $T_{Prop} = 100/3 \cdot 10^8 = 0.3$ μs. This value is clearly negligible as compared to the turn-around and the sensing time of the WiseNET SoC which are both equal to 100 μs (see section 3.3). Let us define the slot[1] time as

$$T_{SLOT} = T_T + T_I \tag{4.5}$$

The length of the vulnerable period around the sensing time is hence of $2T_{SLOT}$. A message will not suffer from a collision if there is no message transmission attempt from neighbors in a period of length $2T_{SLOT}$ centered at the sensing time.

There exist a number of variants of the CSMA protocol. They differ in the actions they take when finding the medium busy during a transmission attempt. Basic variants, introduced in [58], are non-persistent, 1-persistent and p-persistent CSMA.

With non-persistent CSMA, a node having a packet to send and finding the medium busy, re-schedules the transmission to a later time according to a randomly distributed backoff delay B. We will denote with $\bar{B}$ the average backoff delay. A typical choice is to choose for B a uniform distribution between 0 and $2\bar{B}$.

If the nodes of a network can be synchronized, one can use a slotted version of non-persistent

---

[1] The concept of slot will be used when discussing slotted CSMA.

CSMA. The time axis is slotted. Slots[2] have a duration T_{SLOT}. Transmission may only start at the beginning of a slot. At the beginning of a slot, a node is ensured to sense transmissions that have started at the beginning of the previous slot. With slotted carrier sensing protocols, the vulnerability period is halved.

It was observed that with non-persistent CSMA, the channel is often idle while nodes have packets ready to be sent. An attempt to correct this situation and reach a higher channel utilization was made with the proposal of p-persistent CSMA. Let us start with the description of 1-persistent CSMA, which is a special case of p-persistent CSMA. With 1-persistent CSMA, nodes having a packet to send and finding the medium busy wait for the end of the current transmission and start sending as soon as the medium becomes idle. In this sense, they *persist* in their transmission attempt. Of course, if two nodes have a packet to send and wait for the medium to become idle, they will enter in collision with probability 1. If a collision is observed (through a missing acknowledgement), the transmission is re-scheduled after a random waiting time B, as in the non-persistent CSMA case. The idea with p-persistent CSMA is to randomize the transmission attempts at the beginning of the idle period to avoid the probable collisions present with 1-persistent CSMA. P-persistent CSMA is a slotted CSMA protocol. Time slots have a duration T_{SLOT}. The first slot starts when the medium becomes idle. At the beginning of each slot, every node randomly decides either to attempt a transmission (with probability p) or to wait (with probability $1-p$). Once a node has decided to attempt a transmission, it senses the medium. If the medium is found idle, the packet is transmitted. If the medium is found busy, the transmission is re-scheduled after a random waiting time of average value $\overline{B}$.

Since these CSMA variants were introduced in 1975, much research has been devoted to better schedule the transmission attempts and choose the backoff delay distribution. In order to better distribute the transmission attempts, Molle and Kleinrock have proposed virtual time CSMA, where the transmission time of a packet depends on their arrival time [76]. In the IEEE 802.11 standard, it was chosen to decrement a backoff counter after every idle slot and to freeze the counter during busy periods. This scheme permits to distribute randomly the transmission attempts in idle periods and to provide fairness. Indeed, as the backoff counter is not reset when the medium becomes busy, nodes waiting for a longer time will have more chances to transmit than the ones having recently chosen their random waiting delay. In addition, IEEE 802.11 requires nodes to double the backoff window after every collision. This scheme, called binary exponential backoff, is a mean to regulate throughput and avoid congestion.

From the large number of CSMA protocol variants, we have chosen to consider non-persistent CSMA. The reason of this choice are that the different persistent protocols require a node having a packet to send to monitor the channel, and monitoring the channel consumes energy. A node must monitor the channel to detect the end of a busy period, and then to assess that the channel remains idle during a certain time. It can be seen in [58] that optimum p-persistent CSMA provides a lower transmission delay than non-persistent CSMA. However, this difference is small and reached only for the highest traffic loads. In the context of sensor networks, the gain in delay is hence not worth the increase in power consumption.

4.3.3 Non-persistent CSMA with preamble sampling

The combination of non-persistent CSMA and preamble sampling is illustrated in Fig. 4.7. All nodes in a network sample the medium with the same constant period T_W, but their relative

---

[2] Also called a *mini-slot* to emphasize the difference with TDMA slots.

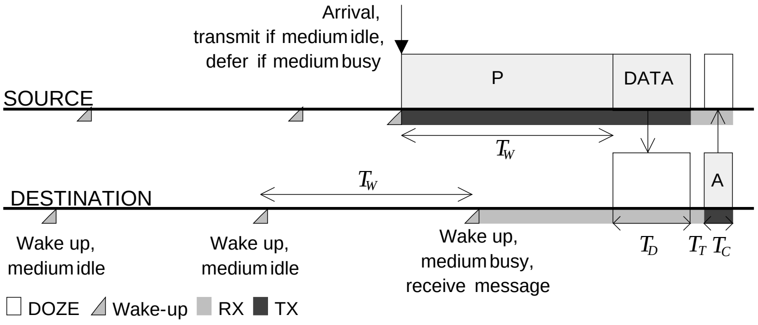

Figure 4.7: Non-persistent CSMA with preamble sampling.

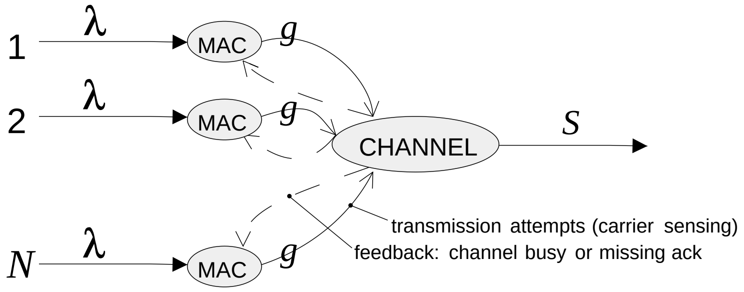
1
λ
MAC
g
2
λ
MAC
g
CHANNEL
S
N
λ
MAC
g
transmission attempts (carrier sensing)
feedback: channel busy or missing ack

Figure 4.8: System model for NP-CSMA analysis.

sampling schedule offsets are independent. A node having a packet to send will precede it with a preamble of length T_W. Before transmitting, a node senses the medium. If the medium is busy, the radio is turned off and the transmission attempt is deferred after a randomly chosen waiting delay of average value $\overline{B}$.

If, during a transmission attempt, a node finds the medium busy, one could be tempted to keep the radio in receive mode to attempt to receive a message. In order to save energy, it is better to turn the transceiver off, and wait for the next scheduled periodic wake-up time, which will be soon enough to catch any transmission.

The performance analysis of NP-CSMA-PS is based on the renewal theory described by Kleinrock and Tobagi in [58]. Their method has been adapted for the analysis of the preamble sampling technique and extended to derive power consumption information. We assume a network of N nodes, where every node is in range of every other node. Poisson traffic with rate λ is generated by every node and addressed to any other node. The sum of the initial transmission attempts and the re-transmission attempts is assumed to be distributed according to a Poisson process of rate g. It is important to note that g is the attempt rate and not the rate with which packets are transmitted. The global attempt rate considering all nodes is Ng. The amount of traffic transmitted with success is measured with the throughput $S \in [0;1]$. As long as the system remains stable, we have $S = N\lambda/T_D$. The system model is illustrated in Fig. 4.8.

4.3.3.1 Throughput

The average throughput can be computed considering the average duration of idle and busy periods. An idle period starts at the end of the transmission of a packet and ends at the start of the next transmission. The duration of the idle period is a random variable $I = T_{SLOT} + X$, where X is the random time between the end of the last transmission and the next arrival. Its cumulative distribution is $\mathsf{P}(X \leq x) = 1 - \mathsf{P}(X > x) = 1 - e^{-Ngx}$, giving a mean of $\mathsf{E}[X] = \frac{1}{Ng}$. Therefore, the mean duration of an idle period is

$$\mathsf{E}[I] = \frac{1}{Ng} + T_{SLOT} \tag{4.6}$$

A busy period starts when the transmission starts and ends when the last interfering packet ends. Its duration is a random variable $Z = Y + T_W + T_D$, where Y is the random time between the start of the first packet and the start of the last interfering packet. To augment the readability of mathematical expressions, the transmission of acknowledgement is neglected. The cumulative probability distribution of Y is $\mathsf{P}(Y \leq y) = \mathsf{P}(\text{no arrival during } T_{SLOT} - y) = 1 - e^{-(N-1)g(T_{SLOT}-y)}$ and its probability density function $f_Y(y) = \frac{d}{dy}\mathsf{P}(Y \leq y) = e^{-(N-1)gT_{SLOT}}\delta(y) + (N-1)ge^{-(N-1)g(T_{SLOT}-y)}$. The mean of Y can be computed as $\mathsf{E}[Y] = T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{(N-1)g}$. The mean busy period duration becomes

$$\mathsf{E}[Z] = T_W + T_D + T_{SLOT} - \frac{1 - e^{-(N-1)gT_{SLOT}}}{(N-1)g} \tag{4.7}$$

The overall throughput of the protocol can be defined as the fraction of the time when the channel carry a data message without collision. If a busy period is successful, it will carry useful information during T_D seconds. A busy period will be successful if the first packet transmitted is the only one, i.e. if no other transmission attempts happen within T_{SLOT}. The probability for a busy period to be useful is hence $e^{-(N-1)gT_{SLOT}}$. The size of the vulnerability period is half of the one considered in Fig. 4.6, because we are interested in the probability of success of any packet, which happens to be the first one, and not of a specific packet. A busy period will be useful during T_D seconds if there is no collision, and during zero seconds in case of a collision. In average, the expected useful time per busy period is $\mathsf{E}[U] = T_D e^{-(N-1)gT_{SLOT}}$ and the throughput

$$S = \frac{\mathsf{E}[U]}{\mathsf{E}[I] + \mathsf{E}[Z]} = \frac{T_D e^{-(N-1)gT_{SLOT}}}{\frac{1}{Ng} + T_W + T_D + 2T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{(N-1)g}} \tag{4.8}$$

4.3.3.2 Delay

A transmission will be successful if none of the $N-1$ neighbors attempt a transmission within the vulnerability period of duration $2T_{SLOT}$ (see Fig. 4.6). The probability of successfully transmitting a packet is hence given by

$$p_S = e^{-2(N-1)gT_{SLOT}} \tag{4.9}$$

A transmission will be attempted immediately at the packet arrival time. If the medium was sensed busy, the node will wait for an average of $\overline{B}$ seconds before to retry. Let b denote the

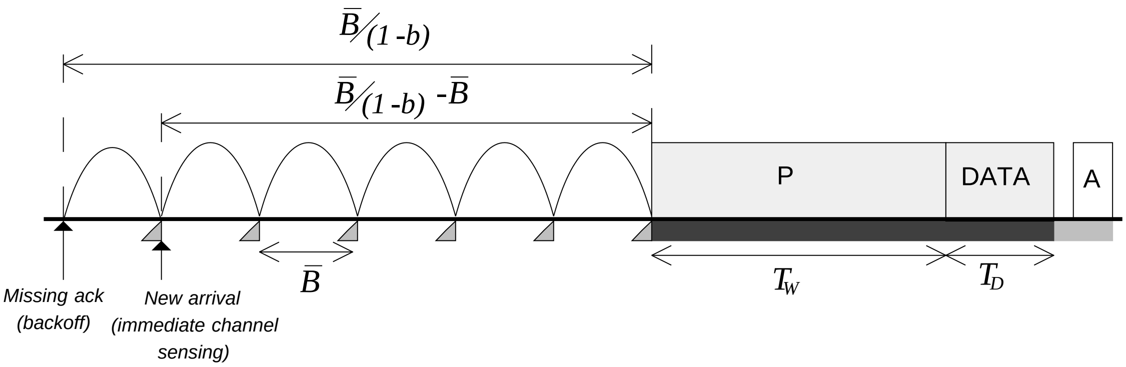

Figure 4.9: Initial waiting delay and waiting delay for retransmissions with NP-CSMA.

busyness of the medium. The medium will be found idle at the first attempt with probability $1-b$. It will be found idle after $k\overline{B}$ seconds with probability $b^k(1-b)$. The average waiting time until the medium is found busy for new packets is hence $\sum_{k=0}^{\infty} b^k(1-b)k\overline{B} = b\overline{B}/(1-b) = \overline{B}/(1-b) - \overline{B}$ (see Fig. 4.9). Once the medium is found idle, the message is transmitted. If the transmission is successful, the additional transmission delay will be $T_W + T_D$. If the transmission was a failure (detected by a missing acknowledgement), the node first waits during the backoff time and then re-attempts the transmission according to the same procedure. In this second case, the waiting delay will be $\overline{B}/(1-b)$ in average.

The probability to transmit a message successfully at the j^{th} transmission is $(1-p_S)^{j-1}p_S$. Let us first assume that nodes backoff at the arrival of new packets. With this assumption, the average duration of the first and the successive attempts would be $\overline{B}/(1-b) + T_W + T_D$. The average delay until the message is transmitted will succeed becomes $(\overline{B}/(1-b) + T_W + T_D)/p_S$. Subtracting the initial backoff delay, one obtains

$$D = (\overline{B}/(1-b) + T_W + T_D)/p_S - \overline{B} \tag{4.10}$$

The busyness of the medium b is obtained from the average length of busy and idle periods. We have

$$b = \frac{\mathsf{E}[Z]}{\mathsf{E}[I] + \mathsf{E}[Z]} = \frac{T_W + T_D + T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{(N-1)g}}{\frac{1}{Ng} + T_W + T_D + 2T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{(N-1)g}} \tag{4.11}$$

4.3.3.3 Power consumption

The power consumption of a sensor node can be computed from the proportion of the time spent in the transmit and receive states. The proportion of the time b_1 when a given node is transmitting is derived as follows: We know that a node is attempting transmissions with a mean rate g. A transmission attempt will result in a transmission only if the medium is found idle, i.e. with probability $(1-b)$. Because the transmissions attempts follow a Poisson process with rate g, the transmissions will follow a Poisson process with rate $g(1-b)$. Following the same method as for the derivation of expression 4.6, one can find that the mean duration of an idle period at the transmitter is given by

$$\mathsf{E}[I_1] = \frac{1}{(1-b)g} + T_{SLOT} \tag{4.12}$$

The mean duration of a busy period at the transmitter being simply the length of a message $\mathsf{E}[Z_1] = T_W + T_D$, we obtain

$$b_1 = \frac{\mathsf{E}[Z_1]}{\mathsf{E}[I_1] + \mathsf{E}[Z_1]} = \frac{T_W + T_D}{\frac{1}{(1-b)g} + T_{SLOT} + T_W + T_D} \tag{4.13}$$

A sensor node periodically samples the medium. If the medium is found busy, it will listen until a message is received. The medium will be busy with transmission from others with a probability $(b - b_1)$. If the medium was found busy, a node will listen to the transmission until the data part is received. In average, the listening duration will be approximately equal to $T_W/2$ (here we assume that $T_W \gg T_D$). As a node samples the medium every T_W seconds, the proportion of the time spent listening is hence given by $(b - b_1)/2$. Using the elements introduced, one can express the mean consumed power as

$$P = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} + b_1 \widehat{P}_T + \frac{(b - b_1)}{2} \widehat{P}_R \tag{4.14}$$

4.3.4 Regular and genie aided NP-CSMA

We will compare the performances of NP-CSMA-PS with those of the regular and genie aided NP-CSMA protocols. The performance parameters of the regular NP-CSMA protocol can be obtained by setting $T_W = 0$ in equations 4.8, 4.10, 4.11, 4.13 and 4.14. We have

$$S^{\mathrm{C}} = \frac{T_D e^{-(N-1)gT_{SLOT}}}{\frac{1}{Ng} + T_D + 2T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{(N-1)g}} \tag{4.15}$$

$$b^{\mathrm{C}} = \frac{T_D + T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{Ng}}{\frac{1}{Ng} + T_D + 2T_{SLOT} - \frac{1-e^{-(N-1)gT_{SLOT}}}{(N-1)g}} \tag{4.16}$$

$$b_1^{\mathrm{C}} = \frac{T_D}{\frac{1}{(1-b^{\mathrm{C}})g} + T_{SLOT} + T_D} \tag{4.17}$$

$$D^{\mathrm{C}} = \frac{T_D + \overline{B}}{(1 - b^{\mathrm{C}})p_S^{\mathrm{C}}} - \overline{B} \tag{4.18}$$

With regular CSMA, a node is either in transmit or in receive state. The mean power consumption is hence

$$P^{\mathrm{C}} = P_Z + b_1^{\mathrm{C}}\widehat{P}_T + (1 - b_1^{\mathrm{C}})\widehat{P}_R \tag{4.19}$$

The preamble sampling technique is a mean to mitigate idle listening. In order to measure the performance of preamble sampling against an absolute benchmark, we introduce the concept of genie aided non-persistent CSMA. In this protocol, as genie tells to each node when the channel is busy. The genie informs the nodes T_S seconds in advance such that they have enough time to power on their radio. A node hence doesn't spend any time listening to an idle channel. This concept helps measuring what could be hopefully approached by feasible methods attempting to replace the genie. Note that this protocol, although helped by a genie, is not yet ideal. It removes the idle listening overhead, but not the overhearing overhead and the collisions.

The mean power consumed by a node using GA-NP-CSMA is the power consumed for reception when the channel is busy because of the transmissions from the neighbors, plus the power consumed when transmitting itself. We have

$$P^{\mathrm{GC}} = P_Z + b_1^{\mathrm{C}}\widehat{P}_T + (b^{\mathrm{C}} - b_1^{\mathrm{C}})\widehat{P}_R \tag{4.20}$$

The other performance characteristics remain unchanged from NP-CSMA.

4.3.5 Performance evaluation

Fig. 4.10 shows the performance of NP-CSMA-PS as a function of the global attempt rate and for three different values of the sampling period (50, 200 and 500 ms). The three first plots show the throughput, the delay and the power consumption. The fourth plot shows the lifetime that can be reached with one AA alkaline battery using the model presented in section 3.2.

It can be observed that the mean power consumption of the regular NP-CSMA protocol cannot decrease below 2.1 mW, the power consumption in receive mode. NP-CSMA-PS can reach a power consumption below 100 μW when the traffic is low. A comparison with GA-NP-CSMA shows the power consumption overhead introduced by preamble sampling. In high traffic conditions, the power consumption of GA-NP-CSMA becomes larger than the one of NP-CSMA-PS. This a-priori astonishing behavior is caused by the fact that, when using NP-CSMA-PS, a node goes back to sleep after having received a packet. It will wake-up only at the next predefined sampling instant, even if the medium becomes busy earlier. With GA-NP-CSMA, a node will wake up as soon as the medium because busy and hence consume more energy. This portion of the curves is anyway of little interest as most packets are lost due to buffer overflow with all three protocols.

The power consumption is best compared when looking at the resulting lifetime in the bottom plot. It can be observed that different sizes for the wake-up preambles are optimum for different amounts of transported traffic.

The throughput is shown in bits per second units, obtained by multiplying S with 25000, the bit rate. This plot shows that the wake-up preambles cause an important reduction in the maximum throughput. In the low traffic region, it can be observed that the throughput grows linearly with the attempt rate. In this region, every transmission attempt result in a transmission, and every transmission is successful.

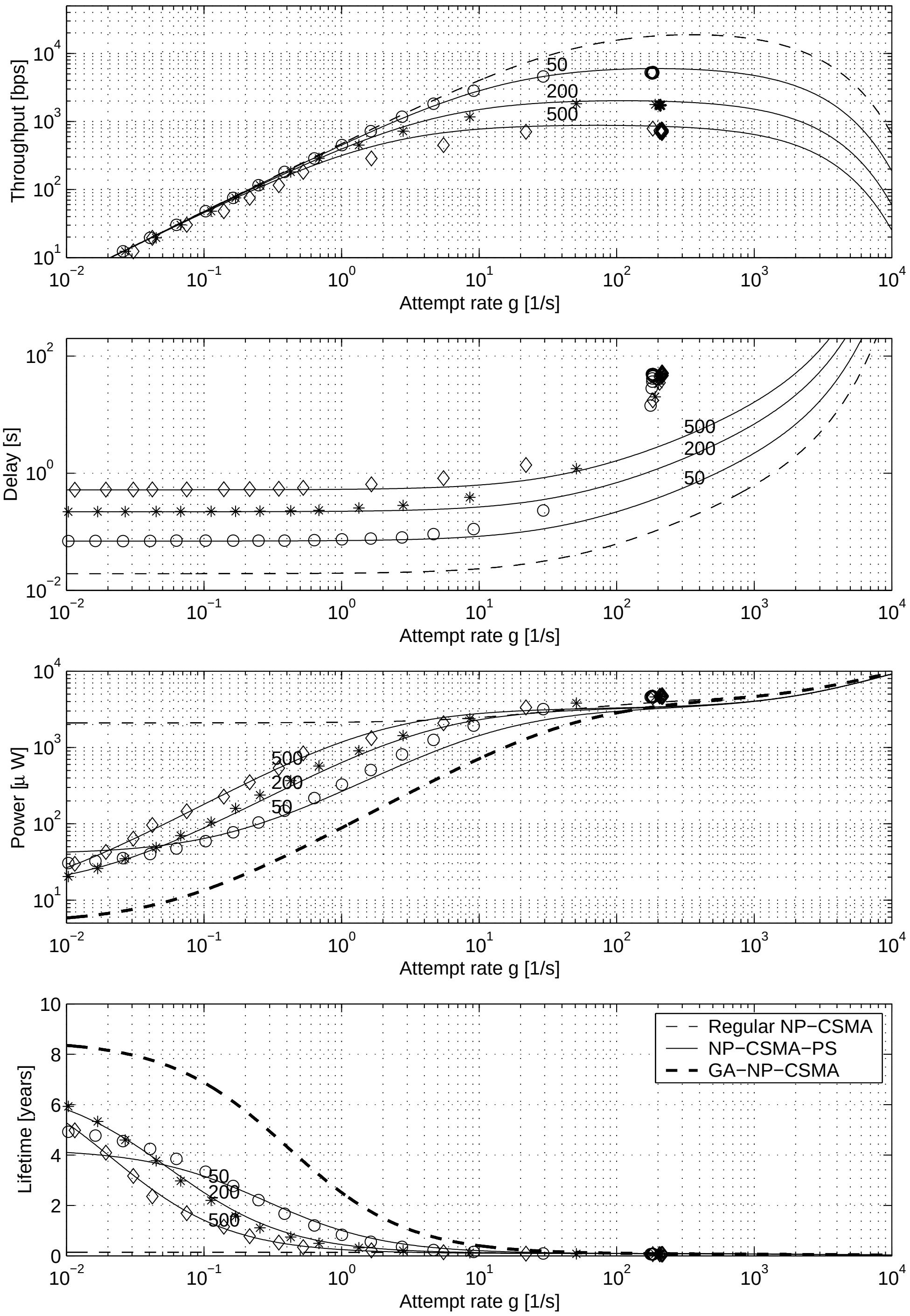

Figure 4.10: Performance of non-persistent CSMA with preamble sampling, as compared to classical NP-CSMA and genie aided NP-CSMA.

In addition to the mathematical analysis, the NP-CSMA-PS protocol was implemented on the GloMoSim platform [134]. This protocol was investigated through simulation first to validate the introduced mathematical model. Second, having simulation results in line with the mathematical model brought confidence in the simulation model, allowing the analysis through simulation of variants of the protocol that cannot be modeled mathematically. In order to simulate low power

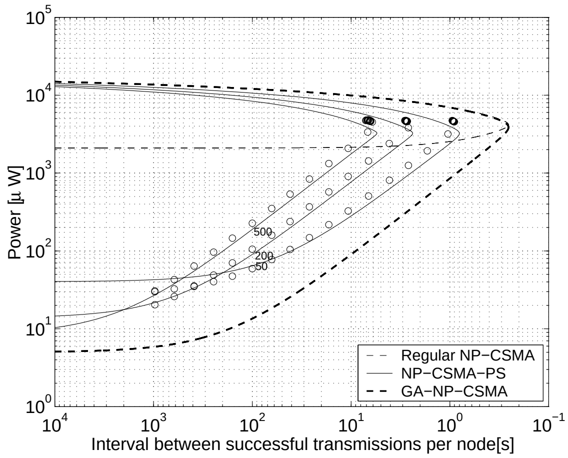

Figure 4.11: Power consumption as a function of the interval between successful transmissions.

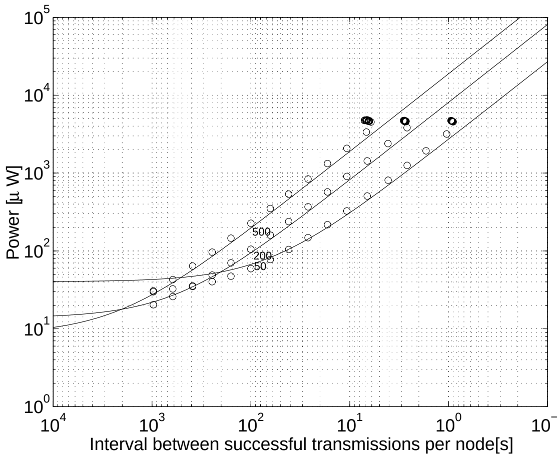

Figure 4.12: Approximation without considering collisions.

protocols, the model of the radio layer has been modified as explained in appendix B. The simulated network consists of 10 nodes numbered 1 to 10 located in range of one another. Traffic is generated following a Poisson process at each node and sent by node i to node $i+3$ modulo 10. A set of simulations is run to measure the performance of the protocol for different values for the injected traffic λ. The attempt rate g used in the mathematical model cannot be chosen as an input for the simulation. In order to display the simulation results together with the theoretical results, the effective attempt rate g was recorded during the simulation, and used for the x-axis coordinates. The markers in Fig. 4.10 show the simulated performance results. In low traffic conditions, every packet generation results in a single transmission attempt. The attempt rate is then equal to the packet generation rate. When the system approach congestion, the attempt rate increases compared to the packet generation rate. With the mathematical model used, the attempt rate can increase without bound. This assumes that every packet is transmitted independently. A node having several packets waiting for transmission would manage a backoff timer for every packet separately. In the simulated model, as would be done in an implementation, the transmission queue is served by a single backoff procedure. The simulated model does hence not display the throughput collapse predicted by the theory.

A more practical representation of the protocol performance is obtained through a plot of the power on the Y-axis and the interval between successful transmissions on the X-axis, as shown in Fig. 4.11. The interval between successful transmissions from a given node $I_{Success}$ is equal to $1/\lambda$ as long as the system is stable. To consider also the congested region, this interval can be obtained from the throughput using $I_{Success} = NT_D/S$.

The average power consumption should remain below 100 μW in order to provide a lifetime of multiple years. This imply that the average traffic must be kept low. With non-bursty traffic, this would mean that the protocol is operated far away from the congestion region. In the low traffic region (up to one packet every 10 seconds per node, up to one packet per second globally), the power consumption of NP-CSMA-PS can be approximated with

$$\tilde{P} = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} + \frac{\widehat{P}_T (T_W + T_D)}{L} + N \frac{\widehat{P}_R (T_W/2 + T_D)}{L} \tag{4.21}$$

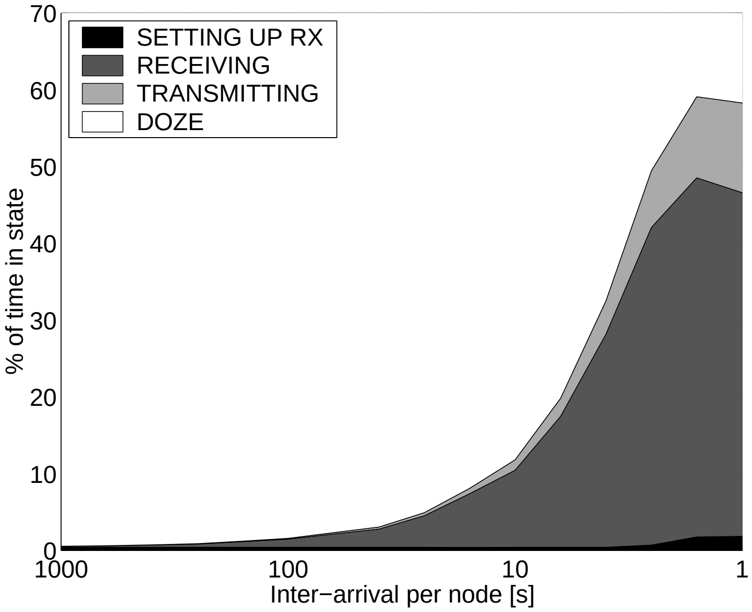

Figure 4.13: Percentage of the time spent by the transceiver its different states using NP-CSMA-PS, as a function of the packet inter-arrival.

This expression includes first P_Z the power consumption in doze state. The second term covers the additional power consumption required to sample the medium every T_W. $\widehat{P}_S T_S$ is the energy to setup the transceiver into receive state, and $\widehat{P}_R T_I$ is the energy to sense the channel. The third term covers the energy to transmit a preamble of length T_W and a message of length T_D every L seconds. The last term represent the receiving cost. The factor N expresses the fact that not only the destination receives a packet, but also all overhearers. As an acknowledgement is much smaller than a wake-up preamble, acknowledgements are neglected. This approximation is compared to the simulation results in Fig. 4.12.

Fig. 4.13 shows the simulated proportion of time spent by the transceiver in its different states using NP-CSMA-PS with $T_W = 200$ ms, as a function of the packet inter-arrival. When the traffic is very low, the dominating element is the time spend in setting up the transceiver into receive mode to sense periodically the channel. With an increasing traffic, the time spend in the receive state soon dominates. With an inter-arrival below 10 seconds, the system enters the congested region. With an inter-arrival below 1 second, the proportion of the time spent in the 'setting up rx' state increases because the frequent transmission attempts require frequent carrier sensing.

The large ratio between the time spent in receive and in transmit mode is caused by overhearing: when a node transmits during $T_W + T_D$ seconds, N nodes listen during $T_W/2 + T_D$ seconds.

4.3.6 Mitigating overhearing

With a wake-up preamble containing only alternating bits, every node surrounding the transmitter must listen during an average of $T_W/2$ seconds before to receive the data packet. If the transmission was a unicast packet, most nodes will have to drop the packet. Solutions to reduce overhearing consist in transmitting information in the wake-up preamble that will allow a node to quickly know if it is the destination of the message. Such solutions include repeating an address information, or repeating the whole data message. The address information could be the full length MAC address of the node. If the address is judged too long, one could transmit a 8-bits hash of the address as proposed in the Hiperlan 1 standard [34].

The information present in the wake-up preamble can also be of use to the intended destination of the transmission. If, together with the repeated address, a counter would indicate the remaining length of the wake-up preamble, the destination could go back to sleep until the data message is starting. If the whole message is repeated, a counter should indicate when the acknowledgement will have to be sent.

Repeating an address presents the advantage of allowing the overhearers to go back to sleep very quickly. Repeating the whole message presents the advantage of adding redundancy. If the first copy of the message is not received correctly by the destination, it will have the possibility to receive one of the following copies. Because of the forward error correction capability brought by the repetition of the whole message in the wake-up preamble, we will consider this second option. Using this optimization, the power consumption of NP-CSMA-PS below the congestion region becomes

$$P = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} + \frac{\widehat{P}_T (T_W + T_D)}{L} + N \frac{\widehat{P}_R (T_D/2 + T_D)}{L} \tag{4.22}$$

The only difference between expressions (4.21) and (4.22) is that T_W is replaced with T_D in the last term. The impact of this optimization is important when the node density N is large and when the ratio between P_T and P_R is small.

4.4 Comparing and combining S-TDMA and NP-CSMA-PS

Fig. 4.14 shows the power consumption of S-TDMA (expression 4.3) and of NP-CSMA-PS with repetition of the data message in the wake-up preamble (expression 4.22) as a function of the transmission interval L. The power consumption of NP-CSMA-PS is drawn for three values of the wake-up period $T_W = 50$, 200 and 500 ms. It can be observed that, for the transport of periodic traffic, S-TDMA is much more energy efficient that NP-CSMA-PS. It must however be kept in mind that this comparison is somehow unfair. We compare a protocol well suited for the transport of periodic traffic and requiring network synchronization, with a protocol well suited for sporadic traffic and not requiring synchronization. The strength of NP-CSMA-PS is in being able to provide a low power consumption when transporting sporadic traffic.

The work presented in this chapter lead us initially in the direction of using spatial TDMA for the transport of frequent data traffic and NP-CSMA-PS for the transport of the supporting signalling traffic. A transceiver capable of providing at least two frequency channels is required[3]. One channel is dedicated for the NP-CSMA-PS contention traffic and one channel is dedicated for the TDMA protocol. If the traffic is small, a single radio should be sufficient to both communicate on the TDMA channel and periodically sample the medium on the NP-CSMA-PS channel. Using (4.3) and (4.22), the power consumption of the combined protocol can be approximated with

---

[3] If more channels are available, a natural extension of Spatial TDMA would be Spatial Frequency and Time Division Multiple Access.

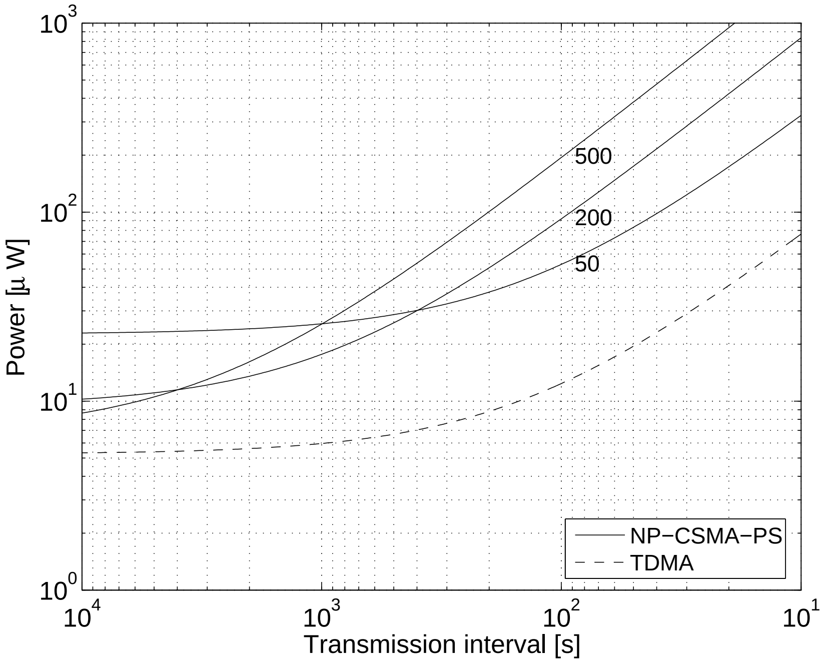

Figure 4.14: Comparison between the power consumption of TDMA and NP-CSMA-PS.

$$\begin{aligned} P &= P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} + \frac{\widehat{P}_T (T_W + T_D)}{L_S} + N \frac{\widehat{P}_R (T_D/2 + T_D)}{L_S} \\ &\quad + \frac{2\widehat{P}_S T_S + \widehat{P}_R (T_D + T_C + 2T_T) + \widehat{P}_T (T_D + T_C)}{L_D} + 4\theta \widehat{P}_R \end{aligned} \tag{4.23}$$

where L_S is the average interval between signalling transmissions on the NP-CSMA-PS channel, and L_D is the period of data transmission on the TDMA channel. This approximation is valid as long as L_S is small enough to keep NP-CSMA-PS out of the congestion region.

Knowing the average data forwarding interval required by the application L_D as well as the average signalling interval L_S required by network management, expression (4.23) can be used to evaluate the lifetime of a node powered with single AA alkaline battery (see Fig. 4.15).

At the initial deployment of a sensor network, it can be expected that a rather extensive signalling traffic will be needed to setup the Spatial-TDMA schedule. This signalling traffic should be transmitted by the application at a rate that will remain below the congestion point. The energy consumption of this initial signalling can be expected to be high, but if the sensor network is operated over a long time, this initial cost will be amortized. Once the Spatial-TDMA schedule initially defined, signalling will be needed to maintain the network, i.e. to repair broken links or insert new nodes.

The design of the energy efficient signalling protocols and algorithms needed for the self-configuration of Spatial-TDMA network is a subject of research by itself. Building on the work presented in this chapter [27], Reason and Rabaey designed and implemented a protocol called On-Demand Spatial TDMA which is using NP-CSMA-PS as a signalling transport mechanism [90].

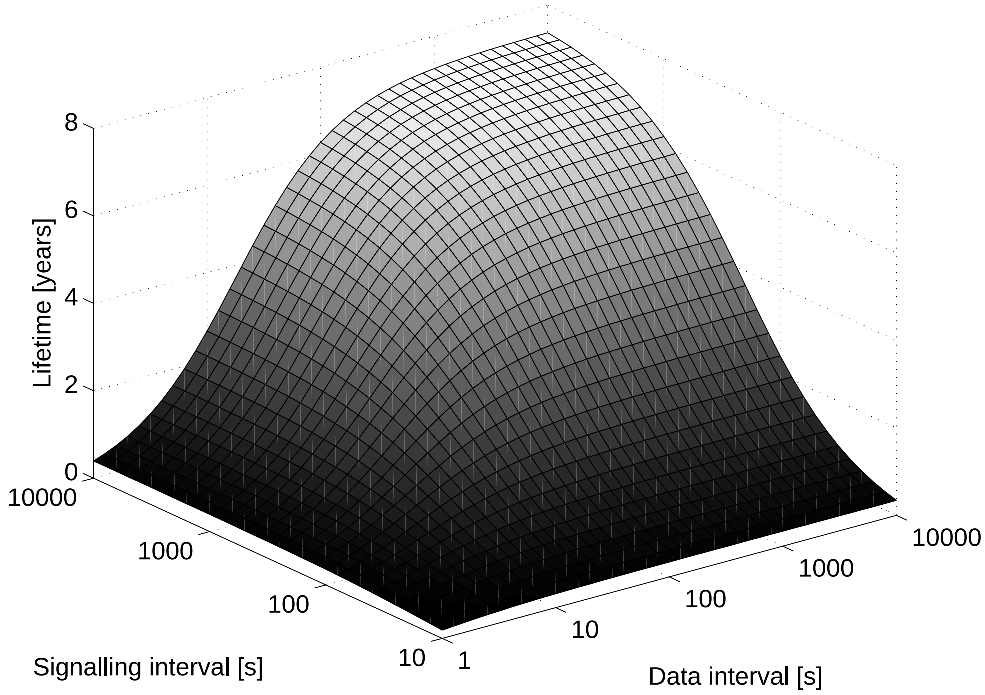

Figure 4.15: Lifetime using Spatial TDMA and NP-CSMA-PS as a function of the data and signalling traffic.

Instead of pursuing this direction of using two protocols separately, we decided to explore an enhancement of NP-CSMA-PS that was expected to approach the energy efficiency of TDMA without introducing the complexity and cost of setting up a TDMA schedule. This protocol will be the subject of chapter 5.

Chapter 5

WiseMAC for Multihop Wireless Sensor Networks

5.1 Introduction

This chapter presents and analyzes a protocol that builds on NP-CSMA-PS to provide both a low average power consumption in low traffic conditions and a high energy efficiency in high traffic conditions. This enhanced protocol has been called WiseMAC (Wireless sensor MAC) after the name of CSEM WiseNET project.

5.2 Protocol description

5.2.1 Overview

WiseMAC is a low power medium access control protocol designed for multi-hop wireless sensor networks. As NP-CSMA-PS, presented in chapter 4, WiseMAC is a contention MAC protocol operating on a single channel using non-persistent carrier sensing (NP-CSMA) for collision avoidance. Idle listening is mitigated using the preamble sampling technique.

The drawback of NP-CSMA-PS is that the wake-up preamble represents a large overhead, leading to a poor energy efficiency. With WiseMAC, acknowledgements carry local synchronization information that is exploited to minimize the length of the wake-up preamble (see section 5.2.2). Systematic collisions that would have been introduced through the synchronization are mitigated using a probabilistic medium reservation scheme (see section 5.2.3). Broadcast and unsynchronized unicast traffic use a random backoff procedure prior to transmission for collision avoidance (see section 5.2.4). Overhearing is mitigated probabilistically when using a short wake-up preamble and through the repetition of data frames within long wake-up preambles (see section 5.2.5). Bursty traffic is transported energy efficiently thank to the ”more” bit, which indicates when more packets are coming (see section 5.2.6). Interruption of data-ack dialogues are avoided through the use of mandatory inter-frame spaces (see section 5.2.7). The receive threshold is chosen well above the sensitivity to maximize useful wake-ups. The carrier sensing threshold is chosen below the receive threshold to mitigate the hidden node effect (see section 5.2.8). The tradeoffs made when choosing the sampling period are finally discussed in section 5.2.9.

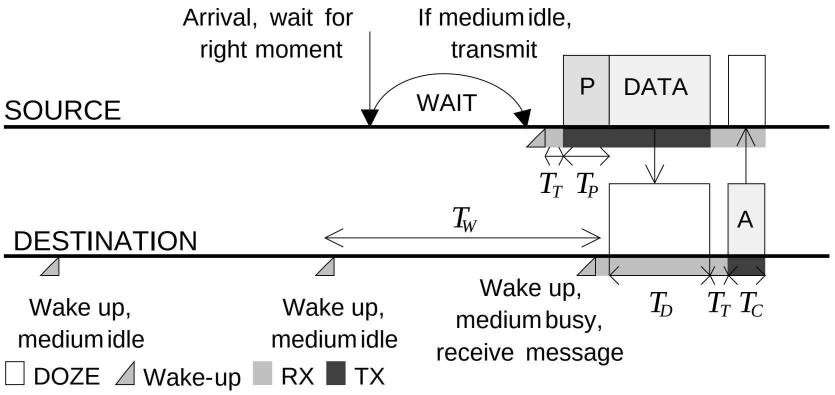

Figure 5.1: Minimizing the wake-up preamble length.

5.2.2 Minimized wake-up preamble

Because the wireless medium is error prone, a link level acknowledgement scheme is required to recover from packet losses. The WiseMAC acknowledgement packets are not only used to carry the acknowledgement for a received data packet, but also to inform the other party of the remaining time until one's next sampling time. In this way, a node can keep a table of sampling time offsets of all its usual destinations up-to-date. Since a node will have only a few direct destinations, such a table is manageable even with very limited memory resources. Using this information, a node transmits a packet just at the right time, with a wake-up preamble of minimized size, as illustrated in Fig. 5.1. In this figure, the duration of the wake-up preamble is denoted with T_P. The wake-up preamble is composed of two parts: the clock drift compensation preamble of duration T_{CDC} and the medium reservation preamble of length T_{MR}. We have $T_P = T_{MR} + T_{CDC}$. The minimum duration of T_{CDC} is addressed in this section. The purpose of the medium reservation preamble is addressed in the next section.

The duration of the wake-up preamble must cover the potential clock drift between clocks at the source and at the destination. This drift is proportional to the time since the last resynchronization (i.e. the last time an acknowledgement was received). Let θ be the frequency tolerance of the time-base quartz and L the interval between communications. As shown below, the required duration of the wake-up preamble is given by

$$T_{CDC} = \min(4\theta L, T_W) \tag{5.1}$$

Expression (5.1) is derived in a similar manner to (4.1): Assume that both source and destination are equipped with a clock based on a quartz with a tolerance of $\pm\theta$. Assume that the source has received fresh timing information from some sensor node at time 0, and that it wants to send a packet to this sensor node at the sampling time L (see Fig. 5.2). If the destination sensor node quartz has a real frequency of $f(1+\theta)$ instead of f, its clock will have an advance of θL at time L. It is hence needed to start the preamble transmission θL in advance. Because the clock of the source might be late, the source must target a time $2\theta L$ in advance to L. Because the clock of the source might be early, and the clock of the destination late, the duration of the wake-up preamble must be $4\theta L$. If L is very large, $4\theta L$ may be larger than the sampling period

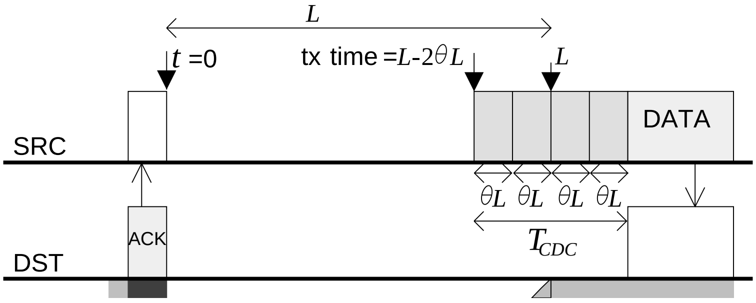

Figure 5.2: Clock drift compensation.

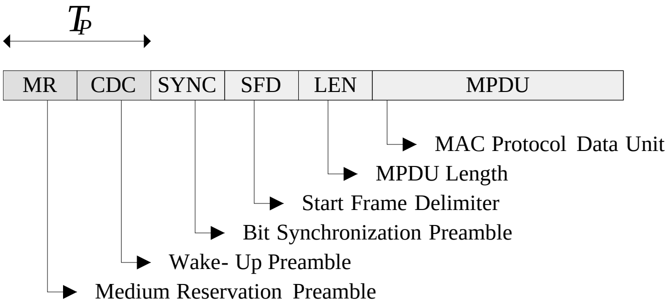

Figure 5.3: Packet format.

T_W. In those cases, a preamble of length T_W is used. We thus obtain $T_{CDC} = \min(4\theta L, T_W)$.

The transmission of the CDC preamble must start at time $L - T_{CDC}/2$, to center the wake-up preamble on the expected scheduled sampling. If the medium is sensed busy at the scheduled transmission time, the attempt is deferred using non-persistent CSMA.

Because the destination of the transmission might theoretically wake up at the very end of the wake-up preamble, it is important not to rely on the wake-up preamble for bit synchronization. The required bit synchronization preamble must be added after the clock drift compensation preamble, as shown in Fig. 5.3.

The first communication between two nodes will always be done using a long wake-up preamble (of length T_W). Once some timing information is acquired, a wake-up preamble of reduced size can be used. The length of the wake-up preamble being proportional to the interval L between communications, it will be small when the traffic is high. This important property, illustrated in Fig. 5.4, makes the WiseMAC protocol adaptive to the traffic. The per-packet overhead decreases with increasing traffic. In low traffic conditions, the per-packet overhead is high, but the average power consumption caused by this overhead is low.

5.2.3 Medium reservation

The synchronization mechanism of WiseMAC can introduce a risk of systematic collision. Indeed, in a sensor network, a tree network topology, with a number of sensors sending data through a multi-hop network to a sink, occurs often. In this situation, many nodes are operating as relays along the path towards the sink. If a number of sensor nodes trie to send a data packet to the same relay, at the same scheduled sampling time and with wake-up preambles of

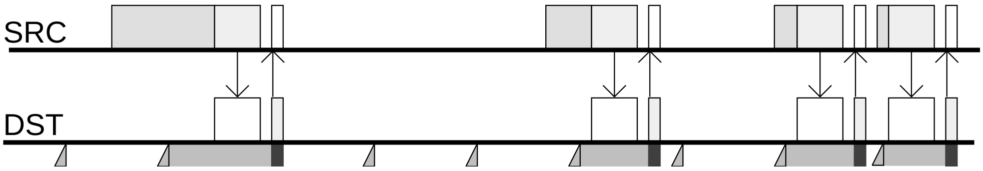

Figure 5.4: Adaptivity of the per-packet overhead to the traffic.

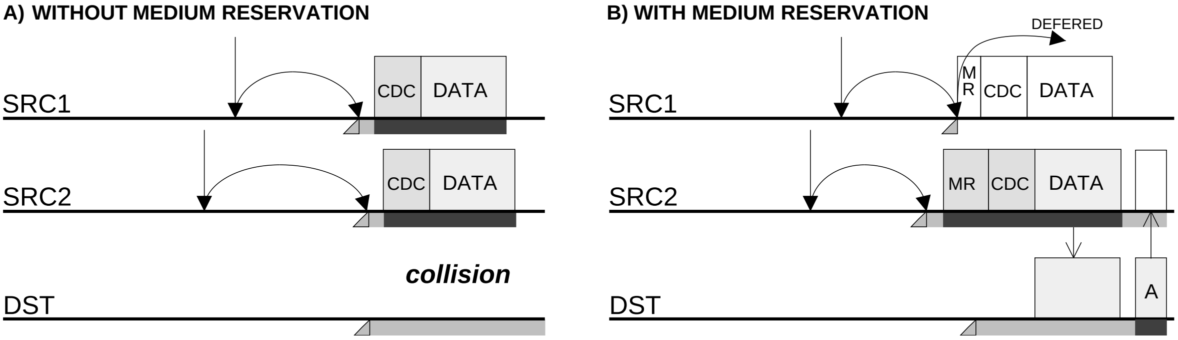

Figure 5.5: Systematic collision between two nodes transmitting to the same destination at the same sampling time (A) and medium reservation (B).

approximately identical sizes, there are high probabilities to obtain a collision.

Assuming that only two stations are attempting a transmission, and that their quartz deviations are θ_1 and θ_2. A guaranteed collision will take place if both stations start their transmission within $T_{SLOT} = 200$ μs seconds (see section 4.3.2). Such condition is met when $|\theta_1 - \theta_2|L < T_T$. In terms of quartz accuracy, the worst case in this context would be to have $\theta_1 \approx \theta_2$. With $|\theta_1 - \theta_2| = 10$ ppm, collisions will occur when $L < 20$ s.

To mitigate such collisions, a solution is to add a medium reservation preamble of randomized length T_{MR} in front of the wake-up preamble. The sensor node that has picked the longest medium reservation preamble will start its transmission sooner, and thereby reserve the medium (See Fig. 5.5).

Because the transmitting nodes are synchronized relatively to the target sampling time, it is possible to use a slotted carrier sensing mechanism. The duration T_{MR} is chosen as an integer random number R multiplied by the slot time T_{SLOT}. The classical choice for the random distribution describing R is to use a uniform distribution in the interval $[0; W_R - 1]$. Let us assume that C nodes are contenting for the medium. Node number 1 has selected a random number r. Nodes $2 \ldots C$ haven chosen r_i, $i = 2 \ldots C$. Let r_{max} be the largest number chosen by these $C-1$ other nodes. The node number 1 will capture the medium and transmit successfully if $r > r_{max}$. It will enter in collision if $r = r_{max}$ and defer its transmission if $r < r_{max}$.

The probability that node 1 has to defer its transmission is given by

$$P_{D_1} = \sum_{r=0}^{W_R-1} \frac{1}{W_R} \left(1 - \left(\frac{r+1}{W_R} \right)^{C-1} \right) \tag{5.2}$$

Proof. $\left(\frac{r+1}{W_R}\right)^{C-1}$ is the probability that all $C-1$ other nodes have chosen a r_i that is smaller than or equal to r. $1-\left(\frac{r+1}{W_R}\right)^{C-1}$ is the probability that one or more other nodes have chosen a larger r_i, causing node number 1 to defer its transmission. This expression is then averaged over all possible values of r. ☐

All other nodes will choose a r_i smaller than r with probability $\frac{r}{W_R}$. In such case, node 1 will capture the medium. The probability to transmit successfully is hence

$$P_{S_1} = \sum_{r=0}^{W_R-1} \frac{1}{W_R}\left(\frac{r}{W_R}\right)^{C-1} \tag{5.3}$$

The probability that node 1 enters in collision can be found using $1 - P_{D_1} - P_{S_1}$. It may also be found directly considering that r is independent of r_{min} and that r is chosen uniformly in $[0; W_R - 1]$. Indeed, whatever the r_{min} resulting from the random selection by the other nodes, a node has one chance over W_R to choose the same number. We hence have

$$P_{C_1} = \frac{1}{W_R} \tag{5.4}$$

Fig. 5.6 shows the probabilities for a node to capture the medium, defer its transmission or enter in collision, for different number of contenders and as a function of the medium reservation window size. Matlab simulations have been run to validate the presented analytical expressions. The simulation results are shown with markers. For $C = 2$, the probability to succeed is equal to the probability to defer the transmission. With increasing C, the probability to defer the transmission increases at the expense of the probability to succeed. The probability that a given nodes suffers from a collision is independent of the number of contenders.

The choice of the medium reservation window size W_R is a trade-off between the power consumption of retransmissions in case of collisions (decreasing with W_R increasing) and the power consumption of transmitting the medium reservation preamble when transmissions are successful (decreasing with W_R decreasing).

In order to be able to select a value for W_R, the energy consumption of a contention resolution between C nodes has been computed through simulations, according to the pseudo code presented in Fig. 5.7. The energy consumption in Fig. 5.7 is computed as follows: If the transmission is a success ($y = 1$) then the energy increment includes, on the source side, the transmission of the medium reservation preamble of size $T_{MR} = r_{max}T_{SLOT}$, of the clock drift compensation preamble $T_{CDC} = 4\theta L$ and of the data message. Finally, on the source side, one must add the energy required to turn-around the transceiver in receive mode and the energy required to receive the acknowledgement. On the receiver side, we have the energy required to receive half the clock drift compensation preamble and the data, the energy required to turn-around and to transmit the acknowledgement.

If the transmission is a failure, the energy at the transmitter is multiplied by the number of nodes y involved in the collision. At the receiver side, the energy required to send the acknowledgement is removed.

The resulting average energy consumption per node (E/C) is shown in the upper plot of Fig. 5.8 as a function of the medium reservation window. The lower plot shows the average number

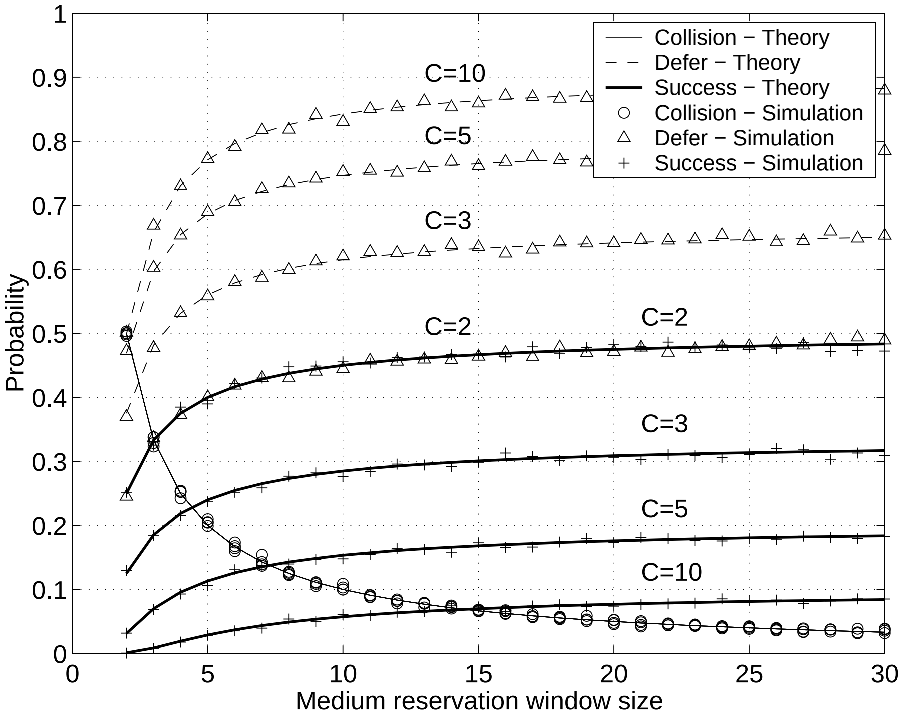

Figure 5.6: Probability for a node to capture the medium, defer its transmission or enter in collision, for different number of contenders and as a function of the medium reservation window size. The markers show simulation results.

```
$x = C$
$E = 0$
$n = 0$
repeat
  $n := n + 1$
  $r_i := \lfloor rand * W_R \rfloor, i \in [1 : x]$
  $r_{max} := \max(r_i), i \in [1 : x]$
  $y :=$ number of nodes having chosen $r_{max}$
  if $y = 1$ then
    $x := x - 1$
    $E := E + \widehat{P}_T(T_{MR} + T_{CDC} + T_D) + \widehat{P}_R(T_T + T_C) + \widehat{P}_R(T_{CDC}/2 + T_D + T_T) + \widehat{P}_T T_C$
  else
    $E := E + y * (\widehat{P}_T(T_{MR} + T_{CDC} + T_D) + \widehat{P}_R(T_T + T_C)) + \widehat{P}_R(T_{CDC}/2 + T_D + T_T)$
  end if
until $x = 0$
```

Figure 5.7: Computation of the energy consumption of a collision resolution.

of contentions per node (n/C). From the number of contentions per node, one can compute the average duration of a contention resolution period, by multiplying it with the number of nodes and the sampling period. The number of collisions per node is simply the number of contentions minus 1. The medium reservation window should be chosen such as to minimize the energy consumption of a collision resolution procedure. The circle markers on the energy consumption

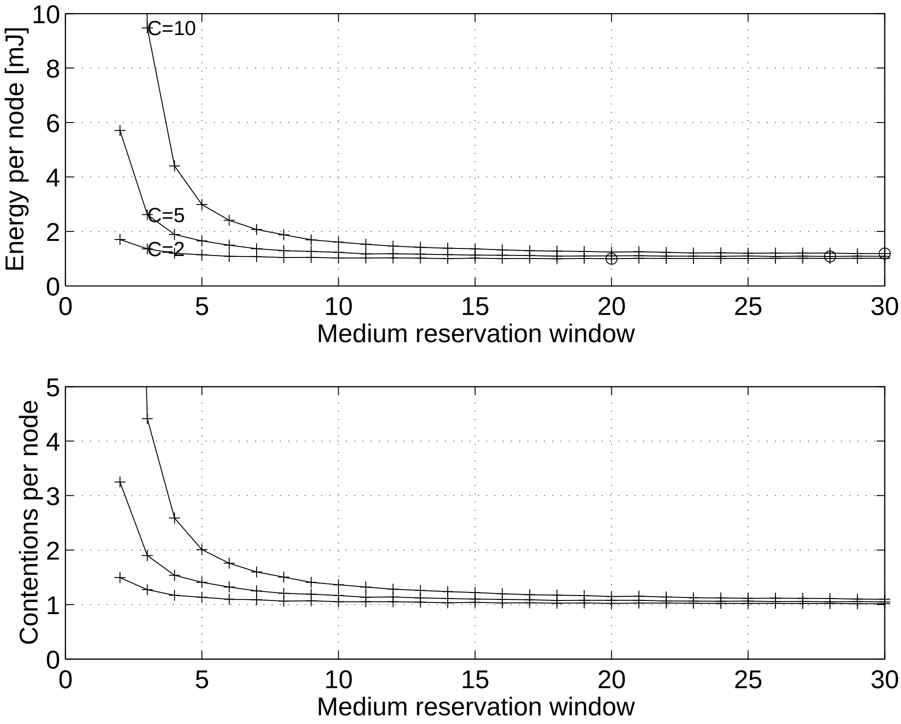

Figure 5.8: Energy consumption per node (upper plot) and number of required contentions per node (lower plot) with a contention resolution between C nodes.

curves show their respective minimum. With $C = 2$, the minimum is reached for $W_R = 20$. With a larger C, the minimum is reached for larger values of W_R, but the difference in energy consumption is very small. Based on Fig. 5.8, we will consider the use of a medium reservation window of length $W_R = 6$. This value is chosen as small as possible to minimize the overhead on traffic that do not need such a contention resolution. With a uniform distribution, the average duration of the medium reservation preamble can be computed using

$$\overline{T}_{MR} = \frac{W_R - 1}{2} T_{SLOT} \tag{5.5}$$

With $W_R = 6$ and $T_{SLOT} = 200$ μs, we have $\overline{T}_{MR} = \frac{W_R - 1}{2} T_{SLOT} = 0.5$ ms, corresponding to an overhead of less than two bytes at 25 kbps.

5.2.4 Random backoff

Collisions between synchronized transmissions towards the same node are mitigated using the medium reservation preamble introduced in last section. Synchronized transmissions towards different nodes are not likely to collide because of the independence among the node sampling offsets. Transmissions using a wake-up preamble of full length T_W are initiated in an unsynchronized way. Such transmissions include broadcasts and transmissions towards a destination for which synchronization information is not available or too old. For such transmissions, collision avoidance will be provided by a random backoff procedure.

The IEEE 802.11 standard [79] specifies that a transmission may be initiated immediately at the arrival of a packet from the upper layer if the medium is found idle, without prior random backoff. A slotted random backoff procedure is used when the medium is found busy. The procedure consists in selecting a random backoff uniformly distributed in $[0; W_B - 1]$ and decrementing the backoff counter by one for every slot that is sensed idle. The backoff procedure is also invoked after a transmission, giving a equal chance to other nodes to seize the channel. If the transmission was successful, the backoff window is reset to its minimum size W_{Bmin}. If the transmission was not successful, the backoff window is doubled up to a maximum of W_{Bmax}. With the DSSS physical layer, we have $W_{Bmin} = 32$ and $W_{Bmax} = 1024$. The exponential increase of the backoff window is a mean to mitigate congestion.

Packet arrivals can be expected to be uncorrelated among nodes in wireless computer networks. This is not always the case in sensor networks. If sensor nodes are programmed to generate an alarm message at the detection of a event, all nodes that have detected some event would attempt a transmission at the same time. Another problematic situation is found with broadcast floods: If sensor nodes are programmed to forward a received broadcast message, all neighbors of the transmitter would attempt a transmission at the same time. For these reasons, it was chosen with WiseMAC to invoke the random backoff procedure before every unsynchronized transmission. With this policy, it is not necessary to invoke the backoff after transmissions.

Because nodes may be synchronized by an external event, it is useful to use a slotted random backoff procedure. The waiting time will be chosen as a random integer number uniformly distributed in interval $[0; W_B - 1]$ multiplied by the slot time T_{SLOT}. The backoff timer is frozen when the preamble sampling activity finds the medium busy, and restarted after the listening phase.

5.2.5 Overhearing mitigation

In PAMAS [103], S-MAC [132] and T-MAC [116], overhearing is mitigated by exploiting the RTS-CTS packet exchange. A node overhearing a RTS or CTS packet turns its transceiver off until the end of the announced transaction.

In WiseMAC, overhearing is naturally mitigated when the traffic is high, thanks to the combined use of the preamble sampling scheme and the minimization of the wake-up preamble length. As already mentioned, sensor nodes are not synchronized among themselves. Their relative sampling schedule offsets are independent. Let T_D be the duration of a data packet. In high traffic conditions, the length of the wake-up preamble T_P becomes small. Let us assume that the total length $T_P + T_D$ of the wake-up preamble and the data packet is then much smaller than the sampling period T_W. As nodes have independent sampling offsets, such short transmissions are likely to fall in between sampling instants of potential overhearers. This intuitive argument is illustrated in Fig. 5.9. A mathematical analysis follows.

When the wake-up preamble is larger than the data message, a further overhearing mitigation mechanism consists in repeating the data message in the wake-up preamble, as illustrated in Fig. 5.10. When data messages are repeated in the wake-up preamble, overhearers can go to sleep after receiving only one copy of the data message. This scheme can also be exploited by the destination of the message to go back to sleep until the acknowledgement transmission time.

Let T_O denote the average duration during which a node overhears a transmission, assuming that the data message is repeated in long preambles. Let T_O^* denote the average overhearing duration when the data message is not repeated. The interest in analyzing T_O^* is to measure the impact of the probabilistic overhearing mitigation alone. The comparison with T_O will give the

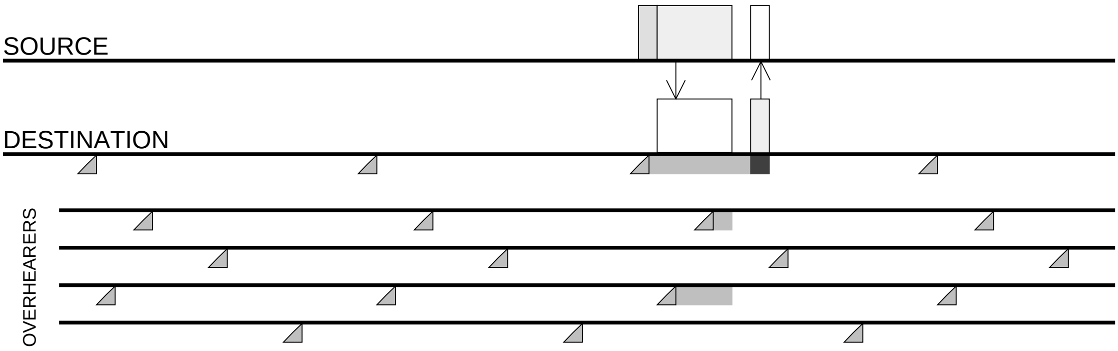

Figure 5.9: Probabilistic overhearing avoidance.

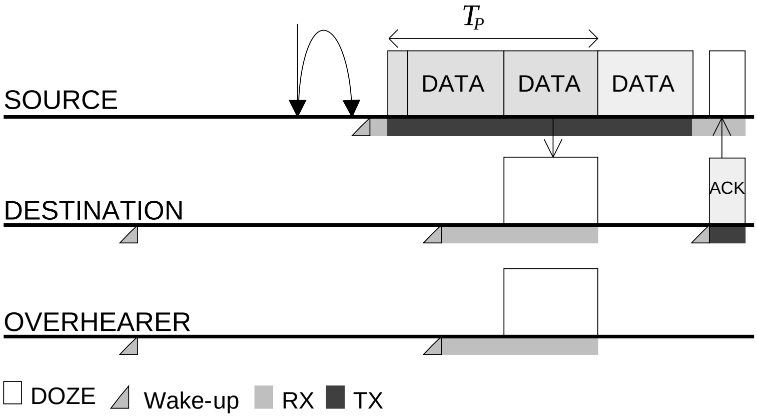

Figure 5.10: Repetition of data message within wake-up preamble.

further gains brought by data repetition. T_O^* and T_O are respectively given by

$$T_O^*(T_P) = \begin{cases} \frac{(T_P+T_D)^2}{2T_W} & , T_P \leq T_W - T_D \\ \frac{T_W}{2} + T_P + T_D - T_W & , T_W - T_D \leq T_P \leq T_W \end{cases} \tag{5.6}$$

$$T_O(T_P) = \begin{cases} (T_P + T_D)^2/2T_W & , T_P \leq T_D \\ (T_D^2 + 3T_D T_P)/2T_W & , T_D \leq T_P \leq T_W - T_D \\ T_D(T_W + 2T_P)/2T_W & , T_W - T_D \leq T_P \leq T_W \end{cases} \tag{5.7}$$

To derive expression (5.6), one must consider the following two cases:

A*) $T_P + T_D \leq T_W$: The length of the transmission is smaller than or equal to the sampling period,

B*) $T_W \leq T_P + T_D \leq T_W + T_D$: The length of the transmission is larger than the sampling period. By design, it maximum length is $T_W + T_D$.

In case A*, assume that a node transmits a packet of duration $T_P + T_D$. An overhearer might sample the medium during this transmission, in which case it will stay awake and listen to

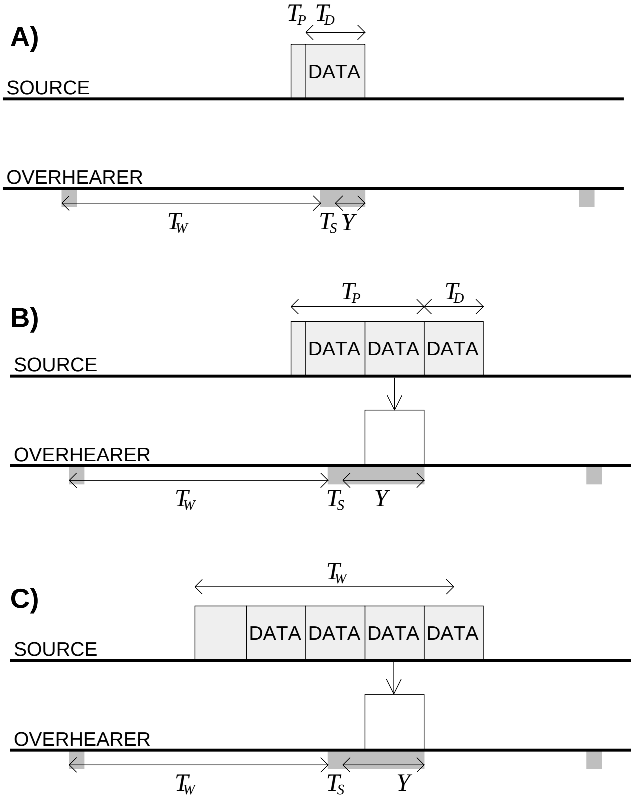

Figure 5.11: Duration of the overhearing period with WiseMAC, when $T_P < T_D$ (A), $T_D \leq T_P < T_W - T_D$ (B) and $T_W - T_D \leq T_P \leq T_W$ (C).

channel until it becomes idle again. Let Y be the duration during which some node overhears a transmission, where $Y \in [0; T_P + T_D]$. The sampling period being T_W, the transmission will not be overheard at all with probability $\frac{T_W - T_P - T_D}{T_W}$, i.e. $\mathsf{P}(Y = 0) = \frac{T_W - T_P - T_D}{T_W}$. With probability $\frac{T_P + T_D}{T_W}$, this node will sample the medium during the transmission, i.e. $\mathsf{P}(Y > 0) = \frac{T_P + T_D}{T_W}$. Y is uniformly distributed over the interval $(0; T_P + T_D]$, we have $\mathsf{P}(Y = y) = \frac{1}{T_W}$ for $Y > 0$. Taking the expectation of the random variable Y, we obtain the average time during which some node overhears a packet not destined to itself $\overline{Y} = \int_0^{T_P + T_D} y \frac{1}{T_W} dy = \frac{(T_P + T_D)^2}{2T_W}$.

In case B*, the overhearer will for sure hit the transmission. If the transmission has a length $T_P + T_D = T_W$, the overhearer will listen to it during an average of $T_W/2$ seconds. If the transmission is longer than that, it will in addition listen to the increment $T_P + T_D - T_W$.

To derive expression (5.7), one must consider the three following cases (see Fig. 5.11):

A) $T_P < T_D$: The wake-up preamble is smaller than a data frame,

B) $T_D \leq T_P < T_W - T_D$: The wake-up preamble is larger than a data frame. The preamble is filled with copies of the data frame. The total length of the packet $T_P + T_D$ is smaller than the sampling period T_W,

C) $T_W - T_D \leq T_P \leq T_W$: The total length of the packet $T_P + T_D$ is larger than the sampling period T_W. By design, T_P must remain smaller than or equal to T_W.

In case A, $\overline{Y}$ is computed identically as in case A*.

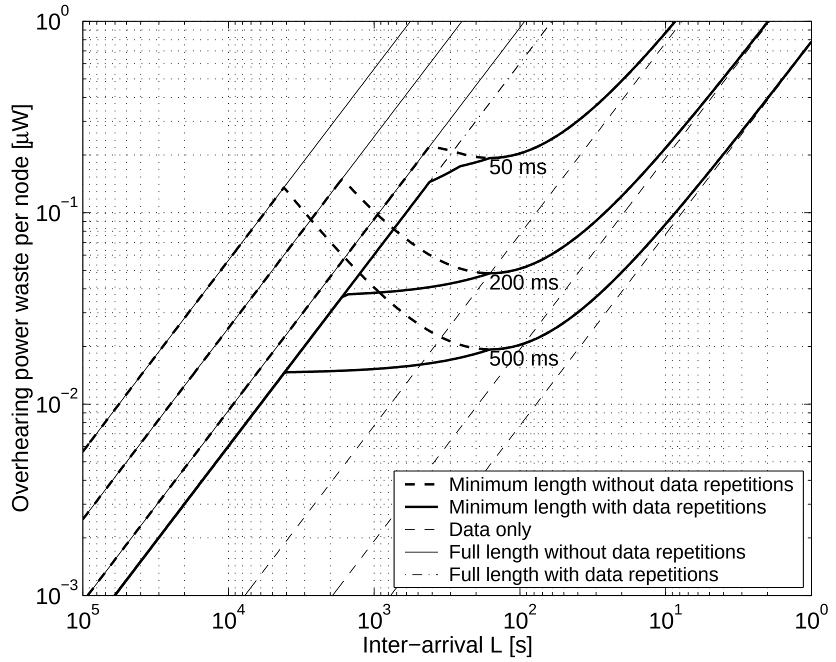

Figure 5.12: Average power wasted by a node overhearing transmissions, as a function of the interval between transmissions ($T_W = 50, 200, 500$ ms, $T_D = 19.2$ ms).

We need to separate case B into three events:

$B1$) The sensor node does not detect the transmission,

$B2$) The sensor node samples the medium during the transmission of the last data frame,

$B3$) The sensor node samples the medium during the wake-up preamble.

Clearly, $\mathsf{P}(B1) = (T_W - T_P - T_D)/T_W$, $\mathsf{P}(B2) = T_D/T_W$ and $\mathsf{P}(B3) = T_P/T_W$. In event B1, the overhearing duration is zero. In event B2, the average overhearing duration is $T_D/2$. In event B3, the average overhearing duration is upper-bounded by $3T_D/2$. This value assumes the reception of half a data frame on average, followed by the reception of one data frame. After the reception of the first complete data frame, the sensor node goes back to sleep. This is an upper bound for two reasons. First, an overhearer can stop listening to the data frame as soon as the destination address field has been found in the header. Secondly, if the sensor node samples the transmission during the initial padding bits, the average delay will be smaller than $T_D/2$ until the first data frame starts. The average duration of the overhearing is hence $\overline{Y}_B = \mathsf{P}(B1) \cdot 0 + \mathsf{P}(B2)T_D/2 + \mathsf{P}(B3)3T_D/2 = (T_D^2 + 3T_DT_P)/2T_W$.

In case C, the duration of the overhearing will always be larger than zero. Let assume that the transmission is longer than T_W. The transmission will be catched in its first $T_W - T_D$ seconds with probability $(T_W - T_D)/T_W$ and will be overheard during an average of $3T_D/2$ seconds. If the transmission is catched in interval $[T_W - T_D; T_W]$, the overhearer will listen during an average of $T_D/2$ plus the remaining of the transmission $T_P + T_D - T_W$. The probability of the second case is T_D/T_W. We have $\overline{Y}_C = \left(\frac{3T_D}{2}\right)\frac{T_W - T_D}{T_W} + \left(\frac{T_D}{2} + T_P + T_D - T_W\right)\frac{T_D}{T_W} = \frac{T_D(T_W + 2T_P)}{2T_W}$. With $T_P = T_W$, this converges to $3T_D/2$.

Fig. 5.12 shows the average power wasted by a node due to the overhearing of transmissions between two other nodes, as a function of the interval between these transmissions. The dashed thick line shows T_O^*, the average power overhead caused by overhearing when data messages are not repeated in the preamble. This overhead is maximized for $L = T_W/4\theta$. This maximum is reached for $L = 416, 1666, 4166$ s for $T_W = 50, 200, 500$ ms respectively. In these cases, it remains below 0.2 μW. The average power overhead is small when L is large, because transmissions causing such an overhead are rare (as L is large). A power overhead of 0.2 μW per node is negligible with a network density corresponding to about 10 nodes within reach of a node. With more neighbors, the overhearing overhead can become problematic. In such cases, the repetition of messages within the wake-up preamble, as illustrated in Fig. 5.10, can be used to further decrease the overhearing overhead. The overhearing overhead T_O when considering the repetition of data message in the preamble is shown with the solid thick lines in Fig. 5.12. With a small sampling period, the gain over the non-repetition scheme is small. The repetition scheme becomes powerful when the ratio T_W/T_D is large. With $T_W = 200$ ms, the power consumption overhead is about 4 times smaller for the worst case traffic $L = 1666$ s. To better understand these curves, Fig. 5.12 also shows the overhearing overhead that would be caused when transmitting the data part of the message alone ($T_P = 0$, thin dashed lines) and when the wake-up preamble length is not minimized (thin solid lines when data messages are not repeated and thin dash-dot line when data messages are repeated in the preamble.).

5.2.6 "More" bit

An important detail of the WiseMAC protocol, which is also found in the IEEE 802.11 power save protocol, is the presence of a *more* bit in the header of data packets. When this bit is set to 1, it indicates that more data packets destined to the same sensor node are waiting in the buffer of the transmitting node. When a data packet is received with the more bit set, the receiving sensor node continues to listen after having sent the acknowledgement. The sender will transmit the following packet right after having received the acknowledgement, without invoking the backoff procedure (see Fig. 5.13). This scheme permits to increase the throughput that can flow through a given forwarder. The maximum throughput, in packets per seconds, is given by the number of buffers available for data frames divided by the sampling period. As long as congestion is not reached on the medium, the throughput is hence limited by the sensor node memory size. Another benefit of the more bit scheme is to reduce the end-to-end delay, especially in the event of traffic bursts.

The *more* bit scheme provides the same functionality as the fragmentation scheme used in S-MAC [132]. An application just needs to segment a large message into smaller packets to obtain the fragmentation behavior. However, the *more* bit scheme is more flexible. Packets that do not belong to the same message but that need to be sent to the same destination will be grouped when using the *more* bit, while they would be sent individually with the fragmentation scheme.

5.2.7 Inter-frame spaces

Between the end of the data message and the start of the acknowledgement, there is an idle period caused by the time needed to turn around the transceiver. The length of this period must be specified to allow inter-operability between transceivers with different turn-around times.

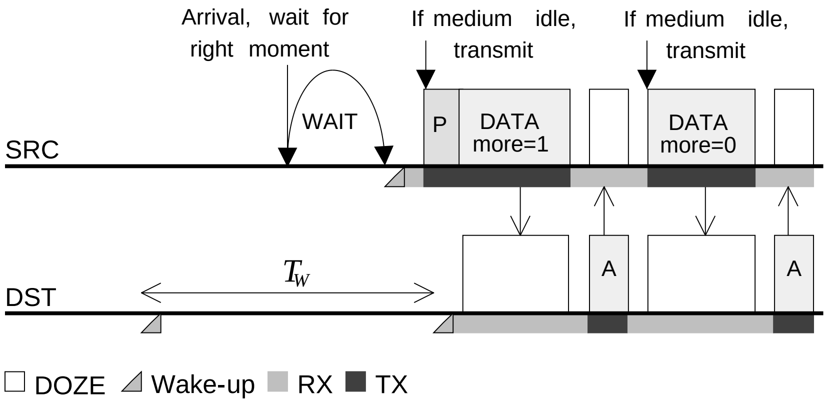

Figure 5.13: Transmission of packet bursts using the "more" bit.

This delay is called $SIFS$ (Short Inter-Frame Space) in the IEEE 802.11 standard. We will use the same terminology. In the IEEE 802.11 standard, $SIFS$ is computed as the sum of the turn-around time, the baseband processing delay and the propagation delay. This precise computation is required because the turn-around time is relatively small ($\leq 5\mu$s with the DSSS physical layer). With the WiseNET SoC, the turn-around time $T_T = 100\ \mu$s is large compared to the baseband processing delay and the propagation delay. For simplicity, we neglect those and use the value

$$T_{SIFS} = 100\ \mu\text{s}. \tag{5.8}$$

If another station attempts a transmission during the period between a data and an acknowledgement packet, it will find the medium idle and initiate the transmission, causing a collision with the acknowledgement message. This problem can easily be avoided by introducing a mandatory waiting time after the end of a busy period, before which any transmission attempt is forbidden. This waiting period is called $DIFS$ (Distributed Inter-Frame Space) in IEEE 802.11. We will use for this interval the value

$$T_{DIFS} = T_{SIFS} + T_{SLOT} = 300\ \mu\text{s}. \tag{5.9}$$

With WiseMAC, as we use non-persistent CSMA, a node does not monitor the medium to find the end of a busy period. To ensure that a data-acknowledgement transaction is not interrupted, a node attempting a transmission and finding the medium idle waits T_{DIFS} and senses the medium again. If the medium is then busy, the transmission attempt is deferred.

5.2.8 Receive and carrier sense thresholds

We have chosen to use a receive threshold that is well above the sensitivity for two reasons. The first reason is that we want to avoid useless wake-ups caused by noise or by weak signals, and wake up only when this is really worth. Here, the lower power consumption is traded against a potential transmission range extension. The second reason is that we use an extended carrier sense range to mitigate the hidden node effect.

The classical approach to mitigate the hidden node effect is to use the request to send (RTS) - clear to send (CTS) dialog before the transmission of the data packet [53, 79]. As the CTS message can be heard by all potential interferers, they remain silent during the data transmission phase. In sensor networks, as the size of data packets is not necessarily larger than signalling messages, the applicability of this approach is questionable. In any case, with WiseMAC, RTS and CTS messages should have a length of T_W seconds to be received by everyone. The resulting large overhead clearly forbids using RTS-CTS reservation messages.

An alternative approach to mitigate the hidden node effect consists in using a carrier sensing range that is larger than the receiver range, such that potential interferers to the destination will be englobed in the sensed area. Let $\mathcal{P}_{CS}$ and $\mathcal{P}_{RMin}$ be respectively the carrier sensing and the receive thresholds.

The received power $\mathcal{P}_R$ at the input of the radio receiver can be computed from the power at the output of the radio transmitter $\mathcal{P}_T$ using

$$\mathcal{P}_R = \mathcal{P}_T - \mathcal{L}_T - \mathcal{L}_P - \mathcal{L}_R \tag{5.10}$$

where $\mathcal{L}_T$ is the loss introduced by the antenna and the radio frequency components on the transmit path, $\mathcal{L}_R$ is the loss introduced by the antenna and the radio frequency components on the receive path and $\mathcal{L}_P$ the path loss. We assume an antenna loss of 2 dB. Such performance is for example achieved with $1/4\lambda$ low cost and small size (2.8 x 1.3 cm) planar antennas from Antenna Factor Inc. [48]. The total losses on the transmit path (caused by the antenna and a spurious emission filter) is assumed to be $\mathcal{L}_T = 5$ dB. Losses on the receive path (caused by the antenna, an antenna switch and an interference filter) is assumed to be of $\mathcal{L}_R = 7$ dB.

For the path loss $\mathcal{L}_P$, we consider the classical one-slope model [100, 67]:

$$\mathcal{L}_P(d) = \mathcal{L}_0 + 10\alpha \log d \tag{5.11}$$

where $\mathcal{L}_0 = 20 \cdot log(4\pi \cdot 868 \cdot 10^6/3 \cdot 10^8) = 31$ dB is the free space loss at 1 meter and α is the power decay index. Using this model, the carrier sensing and receive ranges become

$$d_{CS} = 10^{\frac{\mathcal{P}_T - \mathcal{L}_T - \mathcal{L}_R - \mathcal{L}_0 - \mathcal{P}_{CS}}{10\alpha}} \tag{5.12}$$

and

$$d_R = 10^{\frac{\mathcal{P}_T - \mathcal{L}_T - \mathcal{L}_R - \mathcal{L}_0 - \mathcal{P}_{RMin}}{10\alpha}} \tag{5.13}$$

A signal is assumed to be received without error as long as the ratio between the wanted signal and the sum of interfering signals is above a given SNR. Assuming that the transmitter of the wanted signal is at the receive range d_R, the distance at which a single interference should be located to have the desired signal-to-noise ratio is given by

$$d_I = d_R 10^{\frac{SNR}{10\alpha}} \tag{5.14}$$

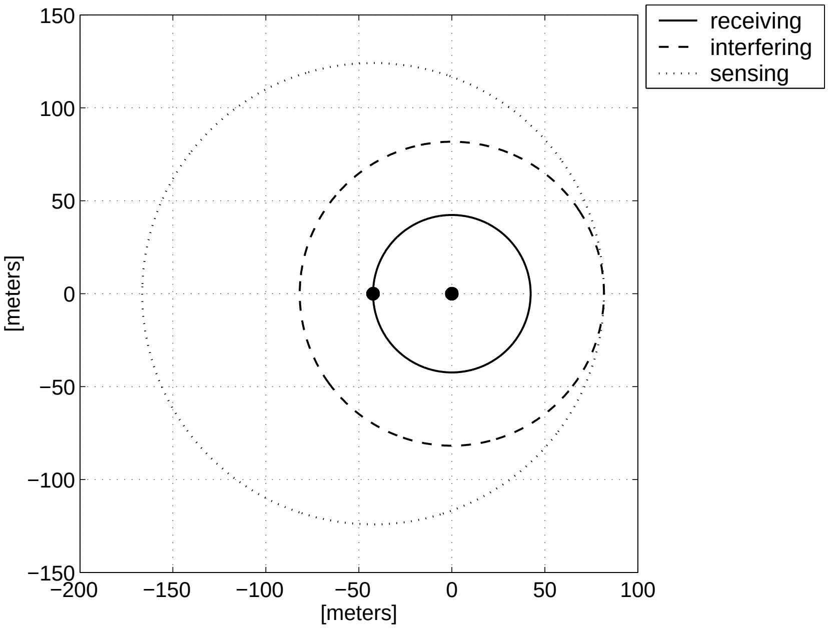

Figure 5.14: Extended carrier sensing range for hidden node effect mitigation.

Assuming that the transmitter is located at distance d_R from the receiver. The potential interferers are located within a circle of radius d_I around the receiver. To ensure that all potential interferers can be sensed by the transmitter, the sensing range must be chosen such that

$$d_{CS} = d_R + d_I \tag{5.15}$$

Using (5.12), (5.13), (5.14) and (5.15), the required receive threshold can be expressed as

$$\mathcal{P}_{RMin} = \mathcal{P}_{CS} + 10\alpha \log\left(1 + 10^{\frac{SNR}{10\alpha}}\right) \tag{5.16}$$

Let us assume that the carrier sense threshold is set at the sensitivity of the WiseNET transceiver $\mathcal{P}_S = -108$ dBm (868 MHz band). With a required SNR of 10 dB and a decay index of 3.5, the resulting receive threshold becomes $\mathcal{P}_{RMin} = -92$ dBm. With a transmit power of $\mathcal{P}_T = 8.5$ dBm and antennas, losses of $\mathcal{L}_A = 2$ dB, the receive, interference and carrier sense range would be of respectively 42, 81 and 124 meters (see Fig. 5.14). The receiving, interfering and sensing ranges is shown in the upper plot of Fig. 5.15 for a varying decay index. The lower plot shows the receive threshold.

Using an extended carrier sensing range increases the exposed node problem: all nodes in the large sensing circle around the transmitting node will remain silent. However, the nodes that are outside the interfering circle could transmit without causing an interference. Intuitively, the exposed node problem seems to lead to a maximum throughput reduction, as nodes that could transmit with success do not transmit. Such a potential maximum throughput reduction was seen as acceptable in the context of sensor networks.

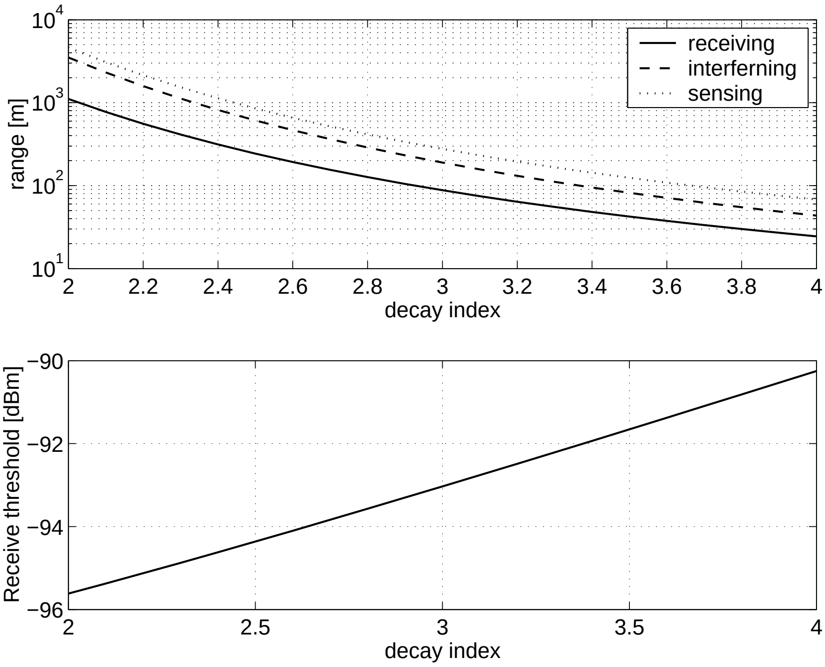

Figure 5.15: Receiving, interfering and sensing range (upper plot) and receive threshold as a function of the decay index (lower plot).

Using an extended carrier sensing range will of course not provide a total protection against the hidden node effect in reality. The main weakness of the extended carrier sensing range scheme is its inability to handle shadowing. If a receiver is in visibility from the transmitter and the interferer, while the interferer is separated from the transmitter by for example a thick concrete wall, the transmitter will be unable to detect signals emitted by the interferer. Secondly, because of multi-path fading, path loss can display large variations over short distances. An additional margin should be included to cover random deviations from the average path loss model. Third, an error can be made on the estimation of the decay index: If the decay index is larger than expected, the protection against hidden nodes will be weakened. Empirical propagation studies have shown that a decay index between 2 and 4 can be expected in an indoor environment [100, 18]. To be on the safe side, it is better to overestimate the decay index. Fig. 5.16 shows the receiving, interfering and sensing ranges that result from estimating the decay index to be 3.5 while it is 3 (left) or 4 (right) in reality. When the decay index is smaller than expected, the sensed region is larger than expected. All hidden nodes are nevertheless covered. If the decay index is larger than expected, some hidden nodes may be located outside of the sensed region.

5.2.9 Sampling period

The choice of the sampling period T_W is a trade-off between the hop delay, the maximum throughput and the average power consumed by the sampling activity (expression (4.4)). T_W should be chosen large enough such that only a fraction of the power budget is consumed by the sampling activity. The larger the value of T_W, the smaller the power consumption of the sampling activity, the larger the hop delay and the smaller the maximum throughput. However, even if the hop delay and the throughput are of little importance to a given application, it is not

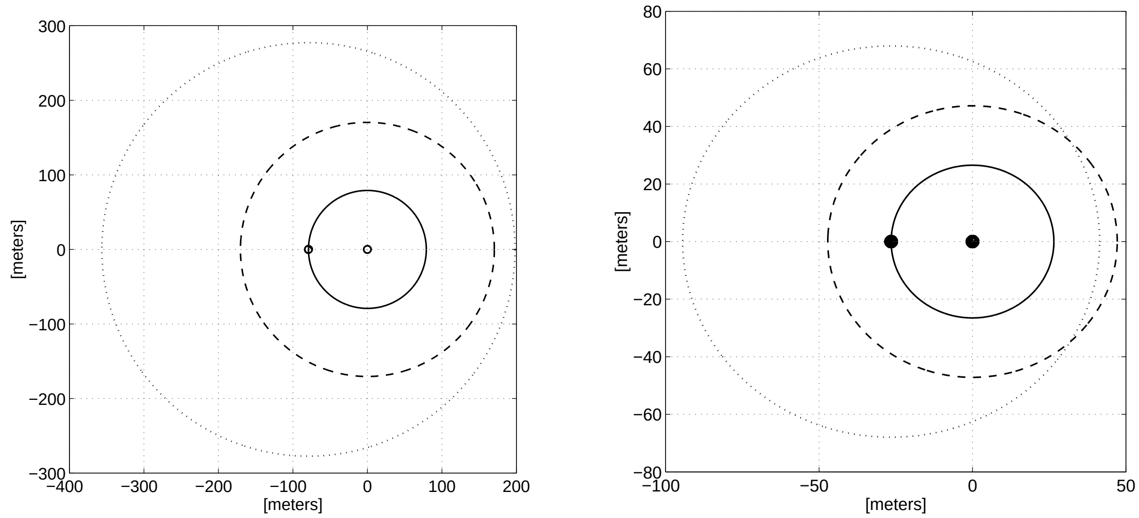

Figure 5.16: Effect of a wrong estimation of the decay index. The estimated decay index used to compute the receive threshold is equal to 3.5 in both cases. The effective decay index is equal to 3 on the left plot, and to 4 and the right plot.

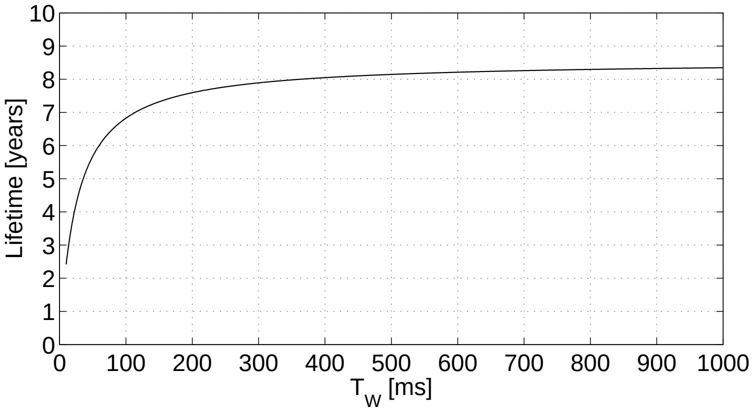

Figure 5.17: Lifetime when sampling the medium with period T_W (no traffic).

interesting, in terms of lifetime, and when using an alkaline battery, to have a power consumption for the sampling activity that is negligible compared to the leaked power. Fig. 5.17 shows the expected lifetime of a sensor node as a function of the sampling period T_W. From this plot, it can be seen that a good value for the sampling period when using the WiseNET transceiver with an alkaline battery is $T_W = 100$ ms, which results in a power consumption of the sampling activity of $(\widehat{P}_S T_S + \widehat{P}_R T_I)/T_W = 8.8$ μW. Using a larger value for T_W increases linearly the delay and decreases linearly the maximum throughput without increasing the lifetime much.

5.3 Performance analysis

The performance of the WiseMAC protocol is analyzed through simulation and theoretical calculations. Comparisons are made with S-MAC, T-MAC, CSMA/CA and an ideal protocol. Models of S-MAC and T-MAC were not available on the used simulation platform. A CSMA

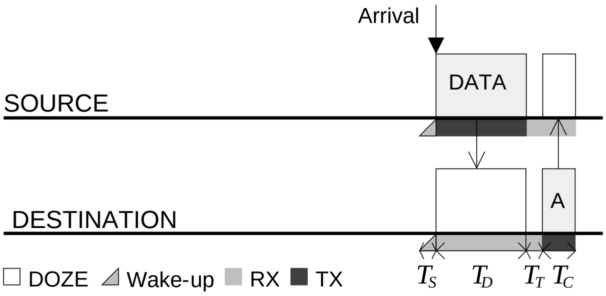

Figure 5.18: Ideal Protocol.

protocol was available but without the RTS/CTS function. In addition to WiseMAC, the S-MAC, T-MAC and CSMA have been modeled on the simulation platform. The S-MAC and CSMA/CA protocols have been implemented as special cases of the T-MAC protocol.

We will consider two scenarios. The first scenario is a network with a regular lattice topology with Poisson traffic flowing in parallel. The second scenario considers a typical sensor network topology with randomly positioned sensors forwarding data towards a sink.

The interest of a lattice topology with traffic flowing in parallel is that it allows exploring the behavior of a MAC protocol without inserting aspects linked to routing, load balancing and traffic aggregation. We focus on idle listening, overhearing and collisions. The regularity of the topology also allows deriving mathematical expressions to approximate the power consumption, and thereby validate the simulation results. Simulations have been also run in a random network topology to evaluate WiseMAC in a more realistic scenario.

The data packet format is the same with all protocols. It is composed of a payload of 46 bytes, a MAC layer overhead of 7 bytes (frame type, source and destination address, sequence number) and a physical layer overhead of 7 bytes (2 bytes bit sync, 2 bytes start frame delimiter, 1 byte for the frame length and 2 bytes for the frame check sequence). The data packet totalizes 60 bytes and has a duration of $T_D = 19.2$ ms at 25 kbit/s.

Acknowledgement packets have a size of 12 bytes (including physical layer overhead) and a duration $T_C = 3.5$ ms (where C refers to control).

5.3.1 Reference protocols

5.3.1.1 Ideal protocol

With the ideal protocol, illustrated in Fig. 5.18, a packet is transmitted over the air as soon as received from the upper layer. The ideal protocol provides hence the lowest possible delay. The destination of the packet ”magically” knows T_S seconds in advance to the transmission start time, that it has to turn on its receiver to receive a packet. Sensor nodes consume energy only to send and receive useful data and acknowledgement packets. There is absolutely no idle listening or overhearing overhead. Real protocols will always consume some energy to implement the wake-up scheme. Their comparison with the ideal protocol will indicate their overhead.

Note that the ideal protocol defined here is better than the genie aided NP-CSMA protocol presented in the previous chapter. With GA-NP-CSMA, the genie did only tell when the channel was busy. With the ideal protocol, a node listens only to its own incoming traffic and hence also avoids overhearing.

5.3.1.2 S-MAC

The S-MAC protocol, already introduced in section 2.3.5, defines sleep intervals, in which all the nodes of the network sleep, and active intervals, in which nodes synchronize and/or can demand to their neighbors to remain awake and receive a transmission [132]. The sum of both intervals is called a frame. The active interval is composed of a phase during which nodes listen for SYNC packets and a phase during which nodes listen for RTS and CTS packets. For fair comparison with WiseMAC, we consider only a phase during which nodes listen for RTS and CTS packets. As SYNC packets are exchanged only sporadically, they may well be transmitted in competition to RTS packets. As illustrated in Fig. 5.19, the listen interval should be long enough for the longest possible backoff followed by an RTS-CTS packet exchange. In the simulation model, the synchronization mechanism has not been implemented. Instead, the network is assumed to be always perfectly synchronized. The model represents an idealized version of S-MAC. The performance results are in this sense an upper bound of what can be obtained with a real implementation of S-MAC that would have to exchange signalling packets to maintain the synchronization.

With S-MAC, one must choose the size of the backoff window as small as possible to minimize the average power consumption. The backoff should however be chosen large enough to avoid collisions with high probability. As proposed in [133], we will use $W_B = 32$. The longest possible contention duration is then $(W_B - 1)T_{SLOT} = 6.2$ ms. After the backoff waiting time, clear channel assessment is repeated two times at an interval of $T_{DIFS} = 0.3$ ms. In the simulation model, RTS and CTS packets have both a size of 12 bytes (including physical overhead) and a duration of $T_{RTS} = T_{CTS} = 3.5$ ms at 25 kbps. The minimum duration of the listen interval is hence chosen to be $T_L = (W_B - 1)T_{SLOT} + T_{DIFS} + T_{RTS} + T_T + T_{CTS} = 31 \cdot 0.2 + 0.3 + 3.5 + 0.1 + 3.5 = 14$ ms. If two nodes exchange successfully an RTS and CTS packet during the listen interval, they remain active until the end of the data and acknowledgement transmissions. Other nodes overhearing these RTS and CTS packets update their virtual carrier sense NAV vector and go to sleep.

A listen time of 115 ms was used on the Mica motes implementation of S-MAC [133]. This is 8 times more that what we selected here. At 10% duty cycle, this results in a frame duration of 1.15 ms which results is quite large hop delays. The reason of this larger listen interval stems from the additional period reserved for synchronization packets and from the lower bit effective bit rate (10 kbps after Manchester decoding on the Mica motes compared to 25 kbps in this simulation). The listen interval chosen here provides much better results and permits a fair comparison of S-MAC with WiseMAC. Note that with a shorter listen interval, the problem of keeping the network sufficiently synchronized becomes more difficult, and that the cost of maintaining the synchronization has been neglected with S-MAC.

Three different frame durations will be considered: $T_F = 1400$, 280 and 140 ms, providing a duty cycle of 1, 5 and 10% in the absence of traffic (10% is the default duty cycle in the implementation of S-MAC on the motes [131]).

5.3.1.3 T-MAC

The T-MAC protocol [116], already introduced in section 2.3.5, is an improved version of S-MAC. The difference is that the duration of the active period is dynamically adapted to the traffic, using a timeout. The active period is ended whenever physical and virtual carrier sensing

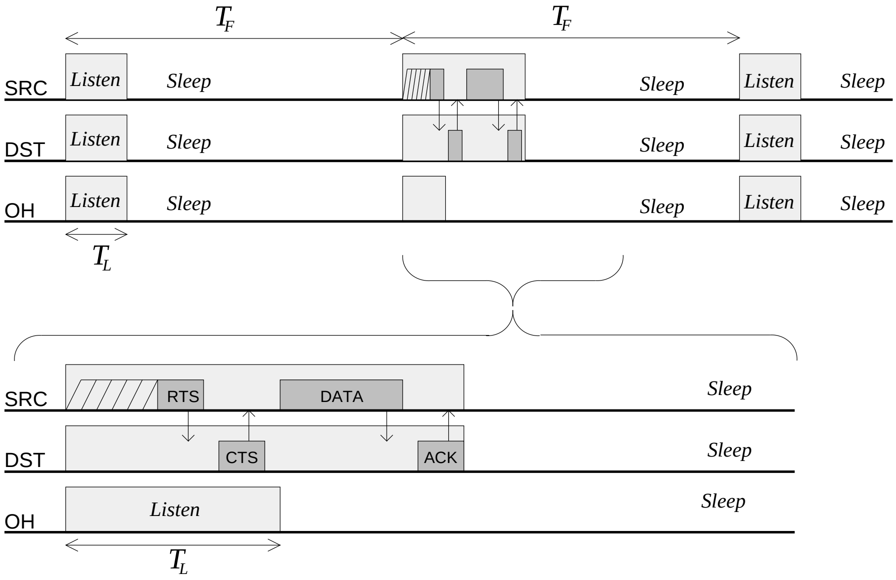

Figure 5.19: Required duration for the listen interval in S-MAC.

find the channel idle for the duration of the time-out.

With T-MAC, one must select the duration of the timeout T_I and the duration of the frame T_F.

As described in [116], the duration of the timeout should allow for the reception of at least the beginning of the CTS packet sent in reply to a RTS packet. It is said that this value shall be larger than $C + T_T + T_{RTS}$ (where C is the maximum duration of the contention) such that the beginning of the CTS message following the RTS message can be received. The authors found that multiplying $C + T_T + T_{RTS}$ with 1.5 gives satisfactory results. In our case, using $W_B = 32$ as in the case of S-MAC, we would obtain $1.5 \cdot (C + T_T + T_{RTS}) = 14.7$ ms. A slightly smaller value $T_I = 14$ ms has been used in the simulations. This gives the time to receive not only the beginning of a CTS packet, but the whole CTS packet.

As with S-MAC, three different frame durations will be considered: $T_F = 1400$, 280 and 140 ms, providing a duty cycle of 1, 5 and 10% in the absence of traffic.

As S-MAC and T-MAC share the same basic simulation model, T-MAC uses data, RTS, CTS and acknowledgement packets of the same size as indicated in the S-MAC section.

5.3.1.4 CSMA/CA

To highlight the power saving introduced by low power protocols, a comparison will be made with the classical CSMA/CA protocol (i.e. using RTS-CTS for hidden node mitigation). With the CSMA/CA protocol, a node is listening to the channel all the time except when transmitting.

This protocol has been modeled based on the T-MAC implementation by setting a duty cycle of 100 %.

5.3.2 Theoretical power consumption

This section will introduce approximate expressions for the power consumption of a relay node, forwarding Poisson traffic with average inter-arrival time L. The node under consideration has

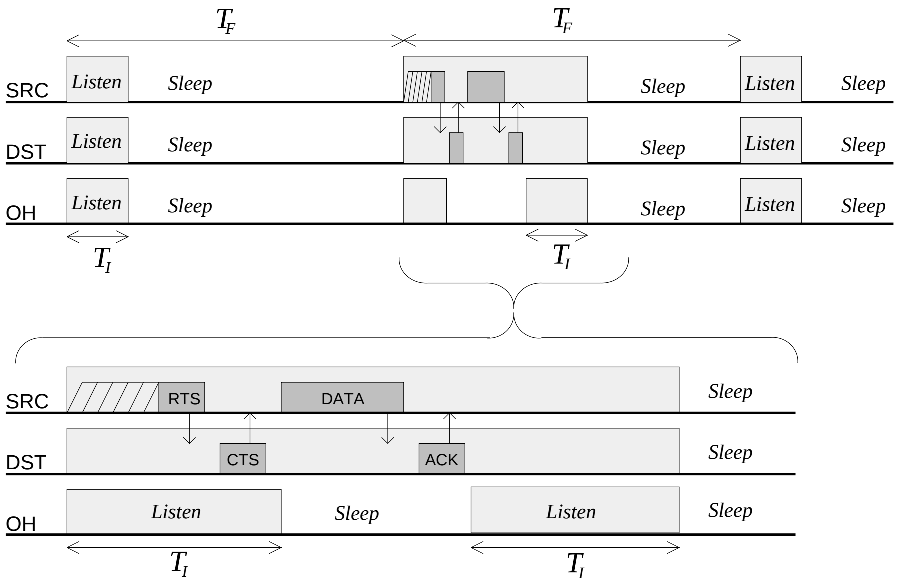

Figure 5.20: T-MAC inactivity timeout.

$N-1$ neighbors that are forwarding traffic as well. Traffic is assumed to be sufficiently low such that collisions can be neglected.

5.3.2.1 Ideal protocol

The power consumed by a relay node forwarding one packet every L seconds using the ideal protocol is

$$\begin{aligned} P_{IDEAL} &= P_Z \\ &+ \frac{\widehat{P}_S T_S + \widehat{P}_R (T_D + T_T) + \widehat{P}_T T_C}{L} \\ &+ \frac{\widehat{P}_S T_S + \widehat{P}_T T_D + \widehat{P}_R (T_T + T_C)}{L} \end{aligned} \tag{5.17}$$

where P_Z, $\widehat{P}_S$, $\widehat{P}_R$, $\widehat{P}_T$ are the power consumption in doze state, the increment in setup state, receive state and transmit state. T_S is the setup time and T_T the turn around time (see the radio transceiver model in section 3.3). The duration of the data and acknowledgement packets are denoted respectively with T_D and T_C. Finally, L is the inter-arrival time.

The second line in expression (5.17) is the power increment required as compared to P_Z to power on, to receive the data packet, to turn-around and transmit the acknowledgement every L seconds. The third term represents the additional power required to power on, to send the data packet, to turn-around and receive an acknowledgement every L seconds (see Fig. 5.18).

5.3.2.2 S-MAC

The power consumption of S-MAC can be approximated with

$$\begin{aligned} P_{SMAC} &= P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_L}{T_F} \\ &+ \frac{(\widehat{P}_T - \widehat{P}_R) T_{CTS} + \widehat{P}_R (T_D + T_T) + \widehat{P}_T T_C}{L} \\ &+ \frac{(\widehat{P}_T - \widehat{P}_R) T_{RTS} + \widehat{P}_T T_D + \widehat{P}_R (T_T + T_C)}{L} \end{aligned} \tag{5.18}$$

where T_F is the frame period and T_L the duration of the listen interval.

This expression is composed first of the power consumed in the doze state and of the power needed to wake-up and listen to the channel during T_L seconds in every frame. These two terms form the basic (traffic independent) power consumption of S-MAC. We have then the power consumed to receive and send a data packet every L seconds.

$(\widehat{P}_T - \widehat{P}_R) T_{CTS} + \widehat{P}_R (T_D + T_T) + \widehat{P}_T T_C$ is the energy required to receive a packet. It includes the energy required to send a CTS, receive a data packet, turn-around and send an acknowledgement. The CTS packet is sent during the listen interval. As the power consumption during the listen interval is already assumed to be $\widehat{P}_R$ in the second term of (5.18), we have to subtract $\widehat{P}_R$ from $\widehat{P}_T$.

$(\widehat{P}_T - \widehat{P}_R) T_{RTS} + \widehat{P}_T T_D + \widehat{P}_R (T_T + T_C)$ is the energy required to send a packet. It includes the energy required to send an RTS, send the data packet, turn around and receive the acknowledgement. Again, $\widehat{P}_R$ is subtracted from $\widehat{P}_T$ as the RTS is sent during the regular listen period.

The power consumption of the signalling required to keep nodes synchronized is neglected.

5.3.2.3 WiseMAC

The power consumption of a relay node with WiseMAC is given by

$$\begin{aligned} P_{WiseMAC} &= P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} \\ &+ \frac{\widehat{P}_T (\overline{T}_{MR} + \overline{T}_{CDC} + T_D) + \widehat{P}_R (T_T + T_C)}{L} \\ &+ \frac{\widehat{P}_R (\overline{T}_{LP} + T_D + T_T) + \widehat{P}_T T_C}{L} \\ &+ \widehat{P}_R (N-1) \frac{\overline{T}_O}{L} \end{aligned} \tag{5.19}$$

This expression is the sum of the power consumed in the doze state, of the the power consumption increments caused by the sampling activity (see expression (4.4)), the transmission of a packet, the reception of a packet and overhearing of this packet by $N-1$ neighbors.

The energy $\widehat{P}_T (\overline{T}_{MR} + \overline{T}_{CDC} + T_D) + \widehat{P}_R (T_T + T_C)$ consumed to transmit a packet includes the energy needed to transmit the medium reservation preamble, the clock drift compensation preamble and the data, as well as the energy needed to turn-around and receive the acknowledgement of duration T_C.

The average duration of the medium reservation preamble can be computed from the slot duration T_{SLOT}, the medium reservation window W_R and the used random distribution. With a uniform distribution, as introduced in (5.5), we have $\overline{T}_{MR} = \frac{W_R-1}{2}T_{SLOT}$.

With periodic traffic of period L, the average duration of the clock drift compensation preamble would simply be $\min(4\theta L, T_W)$. With Poisson traffic of average inter-arrival time L, the average duration of the clock drift compensation preamble must be computed using $\overline{T}_{CDC} = \int_0^\infty \min(4\theta l, T_W)\frac{1}{L}e^{-\frac{l}{L}}dl$ which gives

$$\overline{T}_{CDC} = 4\theta L(1 - e^{-\frac{T_W}{4\theta L}}) \tag{5.20}$$

The energy $\widehat{P}_R(\overline{T}_{LP} + T_D + T_T) + \widehat{P}_T T_C$ consumed to receive the packet includes the energy needed to listen to the wake-up preamble during $\overline{T}_{LP}$ seconds, to listen to the data packet and to send the acknowledgement. Here, we do not count the energy required to setup the transceiver into receive mode, as this energy is already counted for when considering the sampling activity. The medium reservation preamble does not impact the receiver power consumption, as it is supposed to end before the earliest possible sampling time.

To compute $\overline{T}_{LP}$, one must consider separately the cases $T_P < T_D$ and $T_P > T_D$. When $T_P < T_D$, assuming that the clock drift sensor nodes are uniformly distributed in $[-\theta; +\theta]$, the destination sensor node listens on average to the wake-up preamble during $T_P/2$ seconds. When $T_P > T_D$, the data frame is repeated in the preamble. The destination sensor node will start listening during the transmission of some copy of the data frame. It will listen on average during $T_D/2$ seconds before the start of the next data frame. With a periodic traffic with period L, we would have an average listening duration of $\min(4\theta L, T_D/2)$. With Poisson traffic, we have similarly as above,

$$\overline{T}_{LP} = 2\theta L(1 - e^{-\frac{T_D}{4\theta L}}) \tag{5.21}$$

The last term in expression (5.19), representing the overhearing overhead, includes the cost for $N-1$ neighbors to listen during an average of $\overline{T}_O$ seconds. The overhearing overhead can either be seen as the overhead caused by one's transmissions on the other nodes, or the overhead caused by the transmission from other nodes on a given node. If all nodes carry the same traffic, both assumptions lead to the same result.

The average overhearing duration for Poisson traffic with inter-arrival time L can be computed by taking the expectation of T_O (given in (5.7)) over the distribution of T_P. Fig. 5.21 shows a plot of T_O, as a function of T_P. The three curves correspond to the 3 different cases mentioned for the computation of (5.7). The average overhearing duration should be computed by taking the expectation of T_O, a function of $T_P = T_{MR} + \min(4\theta l, T_W)$, with T_{MR} being a uniformly distributed discrete random variable and l being exponentially distributed. It was seen that W_R can be kept small. With $W_R = 6$, the maximum value the medium reservation preamble is $T_{MR} = 1$ ms, which is small compared to a typical message duration. For simplicity, T_{MR} will hence be neglected. To simplify further analytical expressions, we use T_O^B as an approximation of the overhearing in all cases. As can be seen in Fig. 5.21 that the difference with T_O^A is small and the difference with T_O^C is very small, in their respective validity regions. The average overhearing duration can then be computed as $\overline{T}_O = \int_0^{T_W/4\theta} \frac{T_D^2+3T_D4\theta l}{2T_W}\frac{1}{L}e^{-\frac{l}{L}}dl + \int_{T_W/4\theta}^{\infty} \frac{3T_D}{2}\frac{1}{L}e^{-\frac{l}{L}}dl$ which

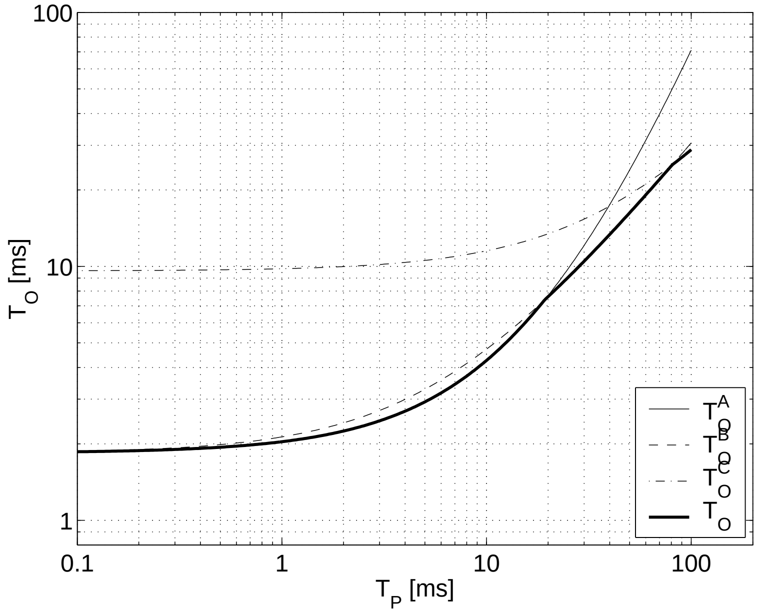

Figure 5.21: Overhearing duration T_O as a function of the preamble duration T_P ($T_W = 100$ ms, $T_D = 19.2$ ms).

gives:

$$\overline{T}_O = \frac{T_D^2 + 12T_D\theta L}{2T_W}\left(1 - e^{-\frac{T_W}{4\theta L}}\right) \tag{5.22}$$

5.3.3 Simulation in a lattice network

5.3.3.1 Topology

In this section, we consider a lattice network topology as illustrated in Fig. 5.22. A separation of 30 meters between nodes is assumed. As in [116], the number of neighbors in range (the node degree) is chosen to be $N = 8$. This number has been chosen small to limit the local traffic but large enough to provide a well connected topology in a random planar network with the same node density (see the percolation theory [39]). Hence, obtained results are also applicable to random plane ad hoc network of equal degree.

5.3.3.2 Traffic

Poisson traffic is generated by nodes on the left (0,9, ..., 72) and transmitted in multi-hop fashion towards nodes on the right (8, 17, .., 80). The rate λ of the packet generation is constant throughout a simulation. As long as no packets are dropped due to congestion, every node in the network forwards packets at rate λ. Simulations results are shown for packet generation rates varying between 0.001 and 1 packet per second (inter-arrival times L between 1000 s and 1 s). Such traffic can be expected in a sensor network for example as a result of a regular data acquisition (e.g. temperature monitoring). Power consumption calculations are done for node

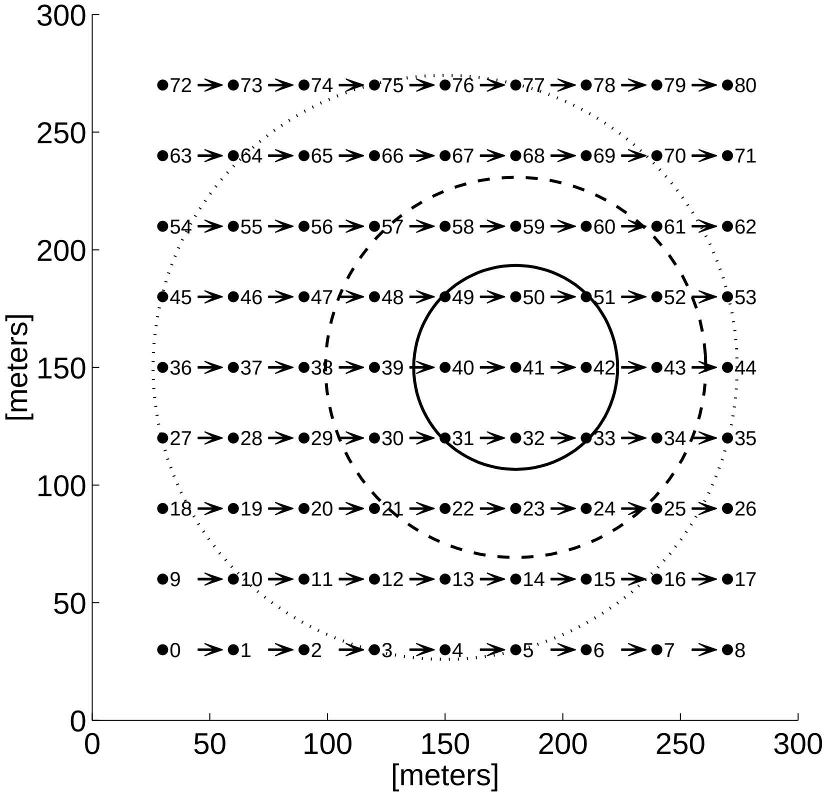

Figure 5.22: Lattice network topology.

number 40. As was shown in [64], the behavior of this central node is approaching the behavior of a forwarding node in a very large network. Having fewer neighbors, nodes on the side of the network will suffer less from overhearing, collisions and backoffs [1]. The duration of the simulation, different for every traffic intensity, has been chosen to amount to $10000 + 200L$ s. This formula permits to have both a sufficient number of transmitted packets in low traffic conditions and a sufficiently large simulated time in high traffic conditions. The simulations produce average values for the power consumption, the transmission delay and the throughput. It could be observed that the used simulation durations were sufficient for the convergence of the observed parameters.

5.3.3.3 Receive, interference and carrier sense ranges

The simulations are run using the path loss model and the receive and carrier sensing thresholds introduced in section 5.2.8. In Fig. 5.22, the solid line circle around node 41 represents the receive range of node 41. All nodes located within this circle may transmit a packet successfully to node 41. The dashed line represents the interference range of node 41 when node 40 is transmitting. The dotted circle around node 40 represents its sensing range. A transmission is initiated by node 40 only if the medium is found idle, corresponding to the situation where none of the 60 other nodes located within the dotted circle is transmitting. A reception is attempted

---

[1] Simulations were also run using cross traffic flowing from the top to the bottom, in addition to the traffic from the left to the right. The obtained power consumption and delay results are slightly increased for all protocols. We will focus on the parallel traffic case, as it is seen as more representative in sensor networks applications.

only if the data is received at a power above the receive threshold.

5.3.3.4 Hop delay

Fig. 5.23 shows the average hop delay. It has been obtained in the simulation by dividing the average end-to-end transmission delay between nodes 36 and 44 by the number of hops (8). The curves are plotted up to an injection rate that causes more than 5% packet loss.

The horizontal dashed line at the bottom of the plot shows the ideal delay $T_D = 19.2$ ms. The delay of CSMA/CA is slightly higher because of the backoff procedure and the exchange of RTS and CTS messages before the transmission of the data.

The delay with WiseMAC is equal to about 120 ms when the traffic is small. It then decreases to a minimum of about 70 ms before to increase again due to congestion. When the inter-arrival time is $L = 1000$ s, the wake-up preamble has a length of $T_P = T_W = 100$ ms. Because the wake-up preamble is maximum, the synchronized transmission scheme is not used. The packet transmission is initiated as soon as the packet has arrived. The transmission delay amounts to the sum of the duration of the preamble and the last copy of the data packet. Here, the delay is measured as the time when the last copy is received. As copies of the data packet are transmitted in the preamble, a smaller delay could be achieved in a single hop transmission when measuring the delay as the time when the first copy is received. In a multi-hop transmission however, the packet cannot be forwarded before the end of the current reception. A node must wait for the reception of the last copy of the data packet in order to send the acknowledgement. Under higher traffic conditions, as the wake-up preamble becomes small, the hop delay becomes shorter. In the worst case, a node has to wait a full period T_W before transmitting the packet to the next node. In the best case, the packet can be transmitted right after having been received. On average, the waiting time is equal to half the sampling period T_W. The average delay can be computed as

$$\overline{D}_{WiseMAC} = T_W/2 + \overline{T}_{MR} + \overline{T}_{CDC} + T_D \tag{5.23}$$

The minimum delay is obtained for $L = 20$ s. At that point, the delay can be computed as the sum of the average waiting time, the length of the medium reservation preamble, the length of the wake-up preamble and the length of the data: $D^{min}_{WiseMAC} = 50 + 1 + 2.4 + 19.2 = 72.6$ ms.

In [133], Ye *et al.* analyze the hop delay with the basic version of S-MAC, and the adaptive one, which corresponds to T-MAC. Their conclusions are in line with the simulation results presented here: With S-MAC, a packet travels one hop during each active period. The hop delay is hence $D^{min}_{SMAC} \approx T_F$. With T-MAC, a packet can travel two hops in every frame, which divides the hop delay by two. We have $D^{min}_{TMAC} \approx T_F/2$. This can be explained as follows: Assume that nodes A, B and C are on a line. Node A sends an RTS to node B. Node B replies with a CTS. Node C, two hops away from node A, overhears the CTS packets sent by node B to node A. This instruct to node C to wake up at the end of the transaction between A and B. Node C will hence be able to receive the packet from node B.

With S-MAC-10% and T-MAC-10%, the frame duration is $T_F = 140$ ms. The average delay resulting from simulation is effectively about 140 ms for S-MAC-10% and 70 ms for T-MAC-10%. In high traffic conditions, the delay increases due to congestion.

On can observe that only T-MAC-10% and CSMA/CA can provide a delay equal to or shorter

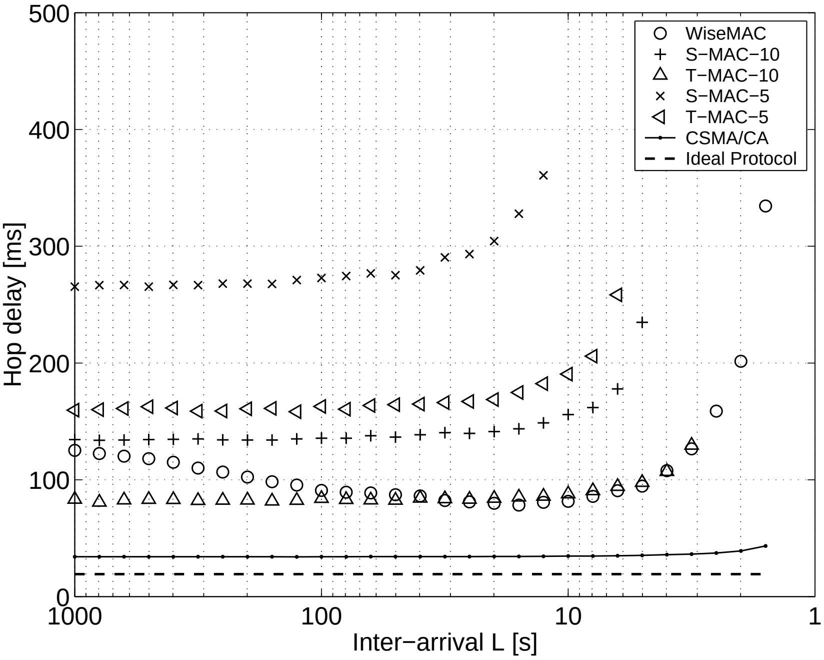

Figure 5.23: Hop delay as a function of the injected traffic (packets have a length of 60 bytes, $T_W = 100$ ms).

than the one provided by WiseMAC. We will see in next subsection that they both consume much more energy.

5.3.3.5 Power consumption

Fig. 5.24 presents the average power consumption as a function of traffic. Power consumption results are collected from simulations by recording the time spent by the radio transceiver of node 40 in its different states (see Fig. B.1 in appendix B). The average power is computed as $\sum_{i \in States} r_i P_i$, where r_i is the proportion of time spent in state i and P_i is the power consumed in that state. This average power corresponds to the task of forwarding one packet every L seconds. In addition to the simulation results, theoretical results introduced in section 5.3.2 are plotted for WiseMAC, NP-CSMA-PS, S-MAC and the ideal protocol.

It can be seen in Fig. 5.24 that WiseMAC consumes a low average power consumption in low traffic conditions ($L > 100$ s). With $L = 100$ s, WiseMAC consumes only 28 μW on average. With a lower traffic, the average power consumption goes below 20 μW. In high traffic conditions ($L < 10$ s), the power consumption of WiseMAC approaches the one of the ideal protocol, which means that WiseMAC achieves a high energy efficiency in high traffic conditions. This property is brought, as already mentioned, by the minimization of the wake-up preamble.

In low and medium traffic conditions, S-MAC and T-MAC consume the same average power. In high traffic condition, the power consumed by T-MAC increases due to the additional time spent by sensor nodes in the receive state because of the timeout scheme. The timeout scheme allows to increase the maximum throughput and divide the delay be two, but causes a small increase in the power consumption as compared to S-MAC for a traffic close to the maximum that S-MAC can transport.

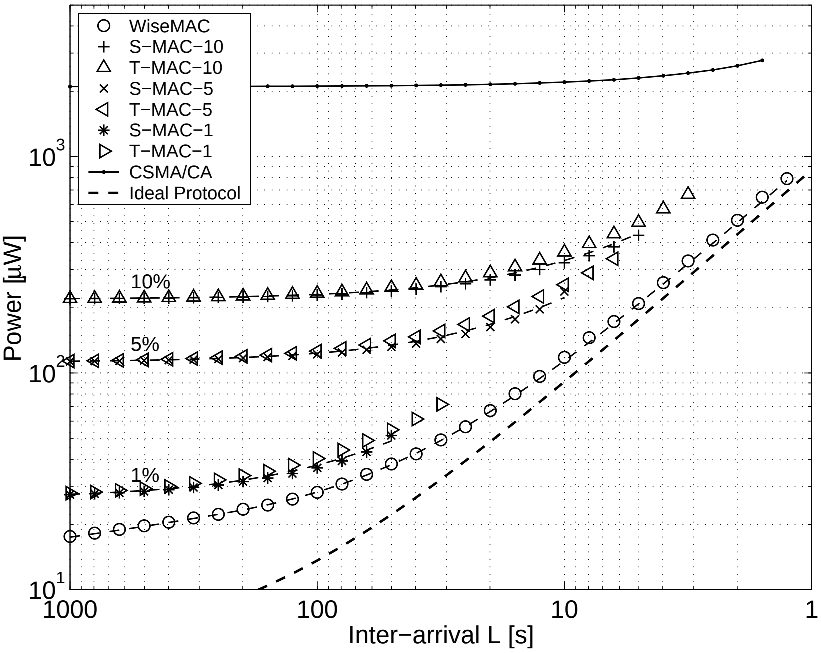

Figure 5.24: Average power consumption as a function of the injected traffic (packets have a length of 60 bytes, $T_W = 100$ ms).

As it does not mitigate idle listening, CSMA/CA consumes a minimum of $P_R = 2.1$ mW, which is 75 times larger than WiseMAC for $L = 100$ s.

Table 5.1 summarizes the power consumed by each protocol for $L = 100$ s, and gives the ratio between their power consumption and the one of WiseMAC.

S-MAC and T-MAC consume at least a fraction of P_R, which corresponds to the selected duty cycle ($0.1 \cdot P_R = 0.21$ mW with 10% duty cycle). The power consumption of S-MAC and T-MAC increases with increasing traffic. With $L = 100$ s, WiseMAC consumes 7 times less than S-MAC or T-MAC at 10% duty cycle. When used at 1 % duty cycle, S-MAC and T-MAC are closer to WiseMAC in terms of power consumption. However, at 1% duty cycle, the frame duration is respectively of $T_F = 1.4$ s. The hop delay is hence of respectively 1.4 s and 0.7 s, i.e. about 14 and 7 times larger than what is provided by WiseMAC. Depending of the choice of the frame duration, S-MAC and T-MAC provide either a low power consumption or a low hop delay. WiseMAC can provide both simultaneously.

Table 5.1: Comparison for $L = 100$ s

Protocol	Power [mW]	Ratio
WiseMAC	0.028	1
S-MAC-10%	0.23	8.1
S-MAC-5%	0.12	4.3
S-MAC-1%	0.036	1.3
T-MAC-10%	0.23	8.3
T-MAC-5%	0.13	4.5
T-MAC-1%	0.039	1.4
CSMA/CA	2.1	75

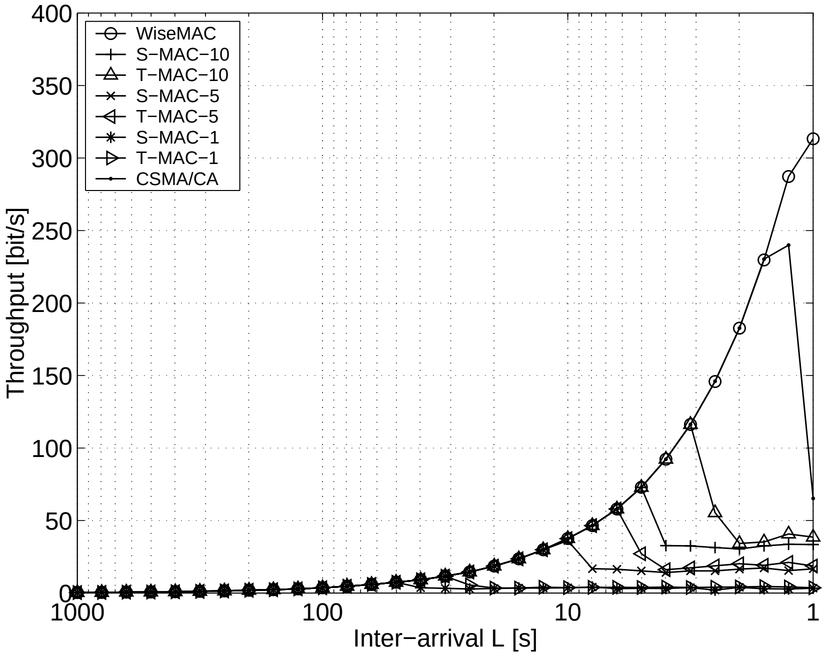

Figure 5.25: Average throughput as a function of the injected traffic (packets payloads have a length of 46 bytes, $T_W = 100$ ms).

5.3.3.6 Throughput

Fig. 5.25 presents the average throughput received by node 44. It has been computed by recording the number of receive packets in the simulation, multiplying this number by the size of the payload (46 bytes) and dividing it by the simulation duration. As the x-axis representing the inter-arrival time is logarithmic, the throughput curve is exponential as long as no packets are lost. Once the congestion region is entered, some packets are lost due to buffer overflows, and the throughput exits this exponential.

The WiseMAC curve enters congestion at about $L = 1.3$ s, which corresponds to an average throughput of 290 bit/s. The reason why the maximum throughput is much lower than the radio raw bit rate (25 kb/s) is to find in the fact that, when using the extended carrier sensing in a multi-hop lattice topology (see Fig. 5.22), only one node from 57 may transmit at the same time. Under this assumption, any CSMA protocol using such an extended carrier sensing is limited to a throughput of $25000/57 = 438$ bit/s.

T-MAC and S-MAC, when operated at 10% duty cycle, enters the congestion region respectively when $L = 3.1$ s and $L = 5$ s, which corresponds to a maximum average throughput of 110 and 70 bit/s. This is respectively 2.5 and 4 times less than the maximum throughput that WiseMAC can transport.

The throughput limitations of each protocol is also visible on the other plots, as simulation results are plotted up to an injection rate that causes more than 5% packet loss.

5.3.3.7 Lifetime

The gain brought by a lower power consumption in low traffic conditions is best visible when looking at the lifetime that can be reached with an AA alkaline battery. Fig. 5.26 shows, using

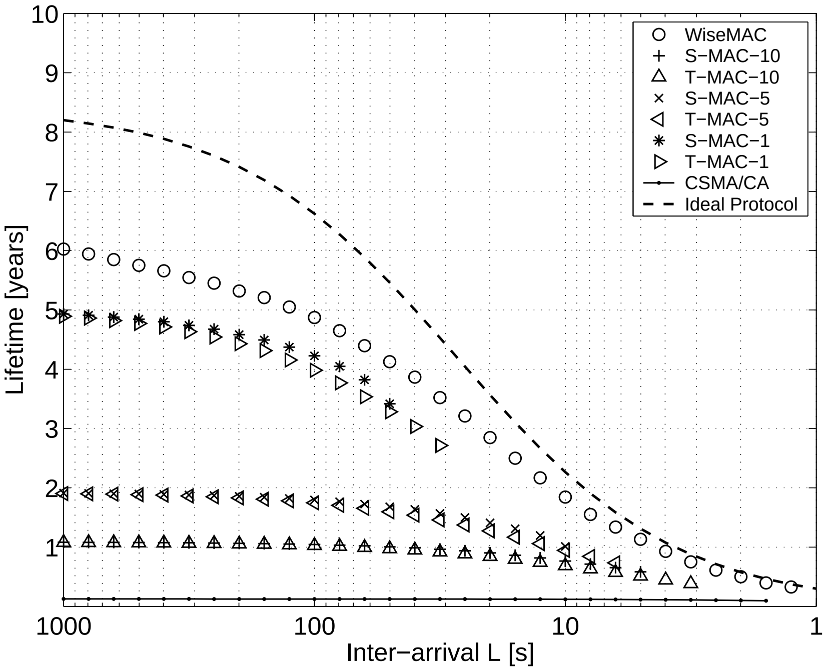

Figure 5.26: Lifetime as a function of the traffic when using a single AA alkaline battery leaking 10% of its initial capacity every year.

the battery model introduced in section 3.2, that a lifetime of five years can be achieved with WiseMAC when forwarding packets at a rate of one every 100 seconds. With S-MAC-10% and T-MAC-10%, a little more than one year is reached.

If the sensor network is operated constantly under a high traffic load (1 packet per second), the lifetime will be very limited with any protocol, even the ideal one. This shows that, if several years of lifetime is a requirement, then high traffic periods should be kept rare. During such periods, it is however important to use a protocol that is very energy efficient. This is the subject of the next section.

5.3.3.8 Energy efficiency

A meaningful metric for the comparison of low power MAC protocols, especially in high traffic conditions, is their energy efficiency. We define the energy efficiency of a MAC protocol as the ratio between the average power consumed by the ideal protocol and the average power consumed by the protocol of interest. The resulting energy efficiency curves for the different protocols are shown in Fig. 5.27.

It can be seen that all protocols have a relatively low energy efficiency in low traffic conditions. Each protocol is associated with a constant minimum power consumption, even in the absence of traffic. With WiseMAC, this minimum overhead is the sampling activity. With S-MAC and T-MAC, it is the cost of listening to the channel during respectively T_L and T_I seconds every period. When the traffic increases, the energy efficiency increases with all protocols, as this basic overhead is shared among more packets. In high traffic conditions, WiseMAC is able to reach a high energy efficiency because the length of the wake-up preamble becomes small when traffic increases. WiseMAC reaches then an energy efficiency above 80%. The energy efficiency

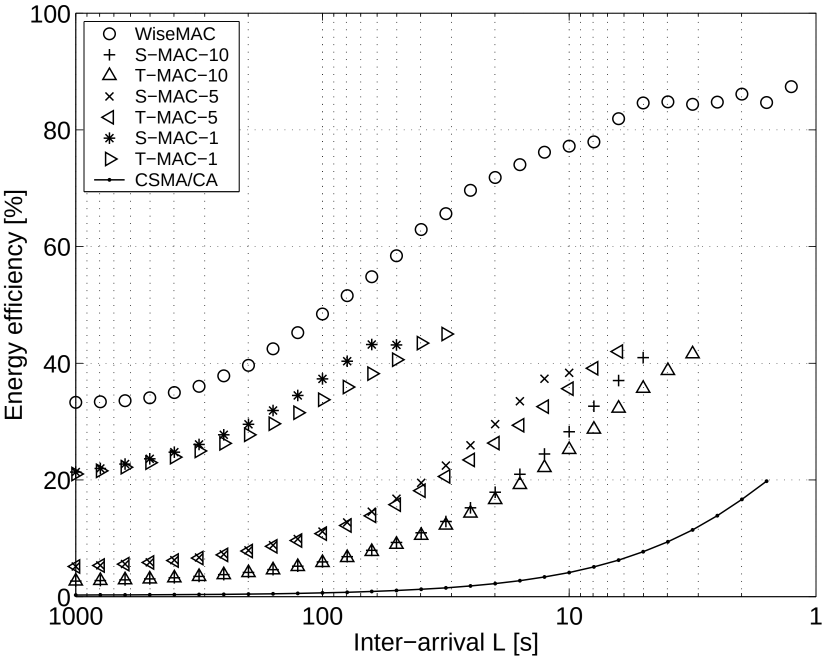

Figure 5.27: Energy efficiency.

of the other protocols remain below 50%.

5.3.4 Simulation in a random network

5.3.4.1 Topology and traffic

Wireless sensor network are often foreseen to operate in a random multi-hop network topology, where sensors forward data to one or more sinks. Such a topology, as illustrated in Fig. 5.28, will be considered in this section. The network is composed of 90 sensor nodes, spread randomly over an area of 300 × 300 meters. Traffic is generated by the 10 black nodes and relayed by the white nodes towards the sink, located on the lower left corner. Routing is pre-computed using Dijkstra's algorithm [20]. The resulting minimum hop routing tree is represented by black lines. The remaining and unused links are represented by gray lines.

The following three experiments will be made:

- **Idle**: No traffic is generated. The simulation is run for 4000 simulated seconds (about 1 hour).
- **Distributed traffic**: The 10 black nodes generate periodically, with a period of 400 s, a packet of 60 bytes. The first node starts at time 0, the second at time 40 s, ..., the last one at time 360 s. Traffic is thus distributed over time. As long as the end-to-end delay remains below 40 s (which will be the case in this experiment), only one packet is in the network at any time. The simulation is run for 4000 s. A total of 100 packets is hence generated.
- **Events**: The black nodes generate periodically, with a period of 400 s, a packet of 60 bytes. They all start at the same times 0, 400, 800, ..., 3600 s. This generate periodically

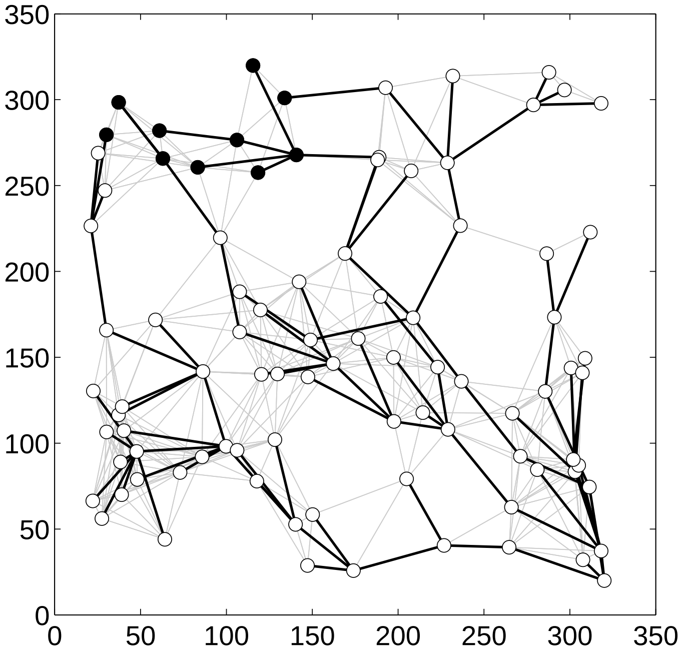

Figure 5.28: Random network topology.

a burst of traffic. Again, the simulation is run for 4000 s and a total of 100 packets is generated.

The purpose of the *distributed traffic* experiment is to explore the behavior of MAC protocols in low traffic conditions. Such a traffic pattern can be expected in many environmental monitoring applications, such as for the periodic measurement of soil moisture in smart agriculture.

The purpose of the *events* experiment is to explore the behavior of MAC protocols in momentary high traffic conditions. Such a traffic pattern can be expected in alarm systems, such as fire or motion detection sensor networks.

In both experiments, a total of 100 packets are forwarded towards the sink. In the *events* experiment, events have been spaced sufficiently such that only 10 packets are in the network at any time. The buffer capacity on each sensor node being of 10 packets, no packets will be lost. Some protocols will require more time to transport the 10 packets than others.

A comparison of the power consumption and delay performances of WiseMAC, S-MAC, T-MAC and CSMA/CA is made in the next sub-section.

5.3.4.2 Power consumption and delay

The bars in Fig. 5.29 show, for the different experiments and MAC protocols, the average power consumption of the nodes. To compute the average power, the total consumed energy is divided by the number of nodes and the simulation time. This average power gives information about the total energy spent in the network. Some node will consume more than others. As the lifetime of a network is often bounded by the lifetime of its weakest nodes, it is important to consider also the maximum average power consumed by any node. It is shown as the "+" markers in Fig. 5.29.

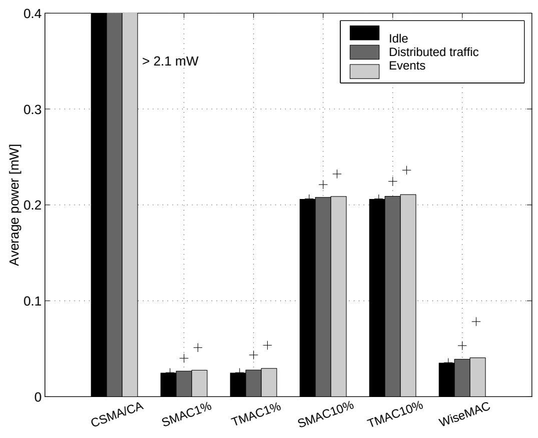

Figure 5.29: Average power consumption.

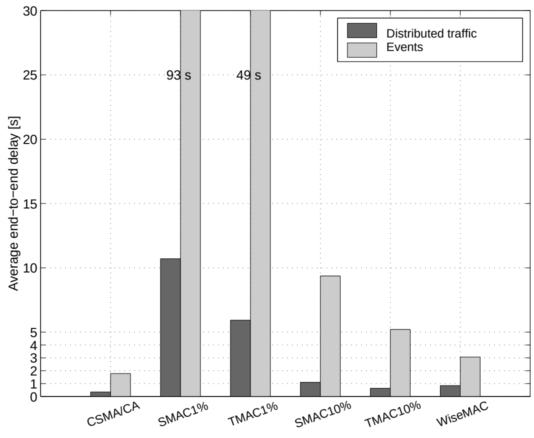

Figure 5.30: Average end-to-end delay.

Fig. 5.30 shows the corresponding average end-to-end transmission delay. This is the average time required by the 100 packets to reach the sink.

The CSMA/CA protocol provides, of course, the lowest average delay for both distributed and events traffic. This is however payed for by a power consumption that is much higher than all other protocols. The power consumption of CSMA/CA is lower bounded by the power consumption in receive mode $P_R = 2.1$ mW.

S-MAC-%1 and T-MAC-%1 provide a low average power consumption, comparable to what is provided by WiseMAC. However, the corresponding delay is very high, while it remains low for WiseMAC. S-MAC-%10 and T-MAC-%10 are able to provide a relatively low delay, but at the expense of a power consumption that is much higher than the one of WiseMAC.

WiseMAC is able to provide both a low average power consumption and a low average transmission delay even in the events experiment. To reach a low average transmission delay with WiseMAC in the event experiment, it is important to use the carrier sensing range extension. Without it, collisions due to the hidden node effect have a large negative impact.

5.4 Sensitivity analysis

5.4.1 Impact of the sampling period

The sampling period of WiseMAC was chosen in section 5.2.9 to be $T_W = 100$ ms, based on the trade-off between the power consumption of the sampling activity and the transmission delay, given that the energy source is a leaking battery. This choice was made without considering any traffic. Figure 5.31 shows the power consumption and the delay that would be obtained with different values of the sampling period as a function of the traffic. It can be observed that, with the parameters of the WiseNET SoC, choosing $T_W = 100$ ms is indeed a good trade-off. With a smaller sampling period, the power consumption in low traffic conditions is notably increased (power is drawn in log scale), while the hop delay is only slightly reduced. Conversely, choosing a larger sampling period only slightly reduces the power consumption at the cost of an important increase in the delay.

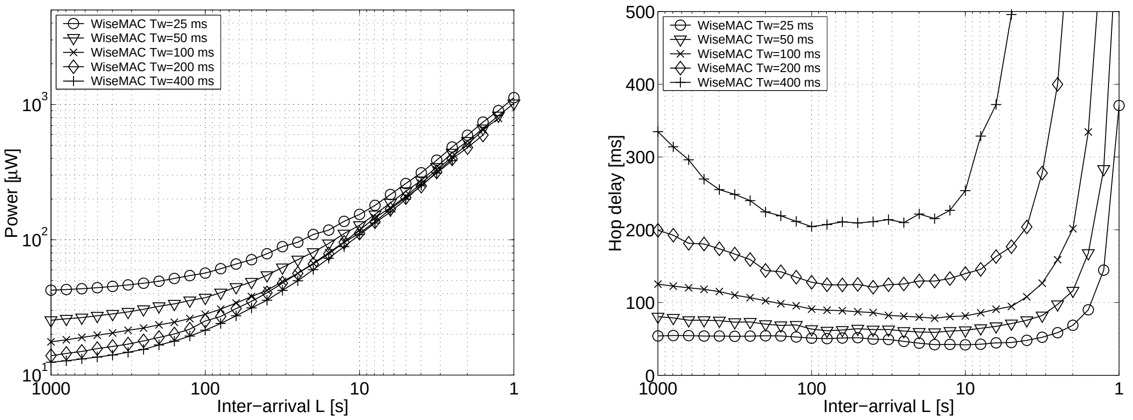

Figure 5.31: Power consumption (left) and delay (right) with WiseMAC as a function of traffic intensity with different sampling periods T_W =50, 100, 200, 500 ms.

5.4.2 Impact of the different schemes used in WiseMAC

A number of schemes have been combined in the WiseMAC protocol. Fig. 5.32 shows the throughput, the delay, the power consumption and the associated lifetime that would be obtained when not using those schemes.

The "no synchronization" curves show the performance obtained when the wake-up preamble is not minimized. This protocol corresponds to NP-CSMA-PS considered in chapter 4. It can be observed that, in medium to high traffic conditions, the gains brought by the preamble minimization are considerable both in terms of maximum throughput and power consumption. In medium traffic conditions (packet inter-arrival times between 10 and 100 seconds) a gain of about 2 years in the lifetime can be observed. In very low traffic conditions (inter-arrival times of 1000 seconds or more), the WiseMAC protocol converge to the NP-CSMA-PS protocol, as synchronization is so inaccurate that a wake-up preamble of the size of the wake-up period must be used.

When looking at the delay, it can be observed that when the preamble is not minimized, the delay is at least equal to the wake-up preamble T_W. Recall that the last copy of the packet determines the transmission delay. With WiseMAC, the waiting time before transmitting a packet varies between 0, if we are lucky and the sampling time of the destination is about to come, and T_W if the sampling time of the destination was just missed. Minimizing the wake-up preamble permits to reduce the average transmission delay to half the sampling period at best.

The "no synchronization and no repetition" curve shows the performance degradation when the data packet is not repeated within long wake-up preambles. The increase in power consumption is rather small. Conversely, the repetition of the data message in long wake-up preambles does not reduce the power consumption much. This is due on the one hand to the large ratio between P_T and P_R with the WiseNET SoC, and on the other hand, to the fact that node density is not very high in the simulated network.

The curve "no extended sensing" shows what would be obtained with a sensing threshold set at the same level as the receive threshold. In low traffic conditions, the performances are identical to the ones of WiseMAC. However, if the inter-arrival time decreases below 10 seconds, collisions due to the hidden node effect and the resulting retransmissions cause an increase of the power consumption. A congestion is observed with $L = 6$ s (60 bit/s) instead of $L = 1.2$ s

(290 bit/s) with WiseMAC, which represent a traffic almost 5 times smaller.

The impact of the "more" bit scheme is mainly in the maximum throughput. This scheme increases the maximum average throughput from 180 to 290 bit/s. In low Poisson traffic conditions, its impact is negligible. This scheme is mainly useful when transporting bursty traffic.

When carrier sensing is done only once, and not repeated after a waiting time of T_{DIFS} seconds, data-acknowledgement dialogues can be interrupted. This effect is visible as soon as the inter-arrival time is below $L = 30$ s through an increase of the power consumption and of the transmission delay. When enforcing a DIFS idle interval before to transmit data, the maximum average throughput grows from 200 to 290 bit/s.

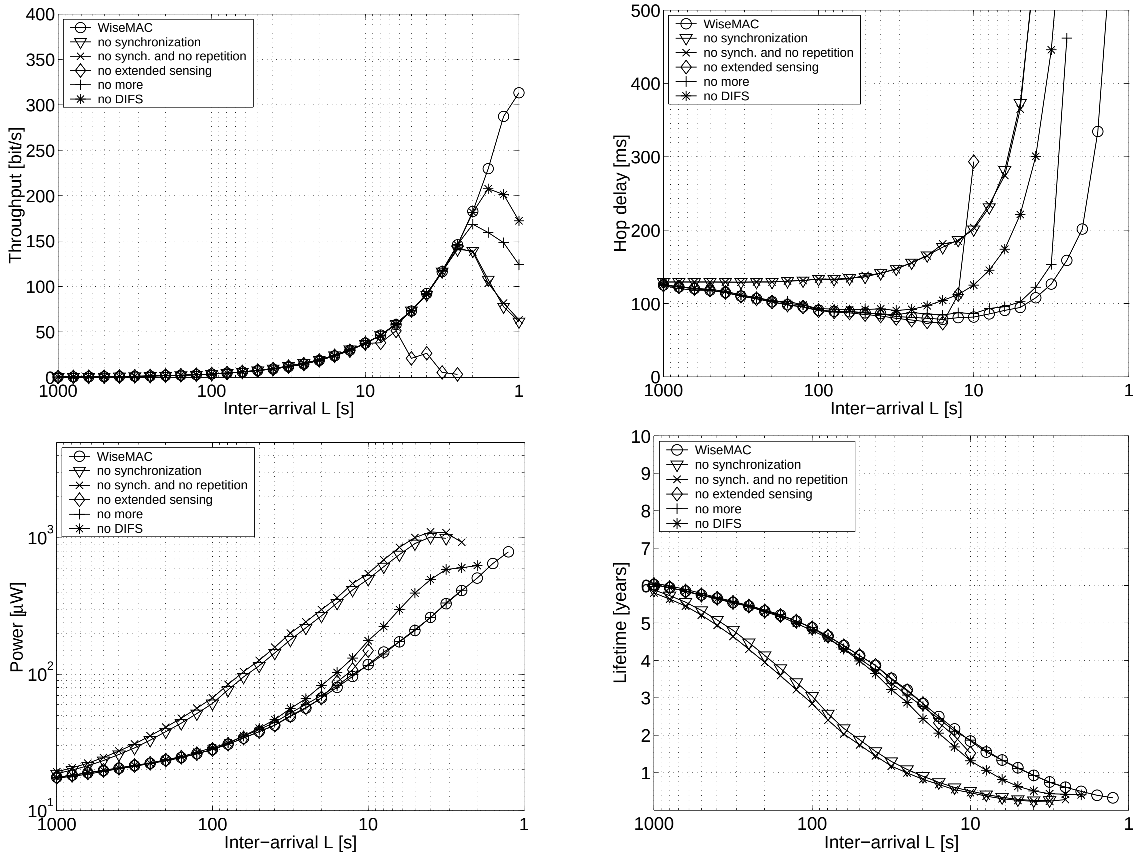

Figure 5.32: Throughput (top, left), delay (top, right), power consumption (bottom, left) and lifetime (bottom, right) with WiseMAC as a function of traffic intensity without one of the following schemes: Wake-up preamble minimization, extended carrier sensing, more bit, mandatory inter-frame space before transmission.

5.4.3 Impact of external interferences

Interferences from other systems can be frequent, depending on the chosen frequency bands. In the ISM bands, interference of long duration can be expected. During such phases, no communication will be possible. In addition, the presence of interferences has an impact on the wake-up scheme of WiseMAC. With the WiseMAC protocol, a sensor node keeps listening when it samples the medium busy. If the medium was found busy because of an interference, the sensor node will loose energy trying to receive a non-existent packet. It is very important to include in the WiseMAC protocol a mechanism to differentiate a wake-up preamble from an interference. We have chosen to repeat the data frame in the preamble, when the preamble is larger than a data frame. In a valid transmission, a data frame shall hence be received within at most $2T_D$ seconds. Interfering signals can hence be recognized using a timeout.

5.4.4 Importance of the transceiver parameters

For low duty cycle applications, as the transceiver is frequently turned on and off, it is of the highest importance to minimize the energy required to turn the transceiver on. The power consumed in receive mode should be very low for two reasons. Firstly, the power consumed in receive mode is related to the setup energy, and secondly, depending on the used MAC protocol, much energy may be wasted through idle listening or overhearing. The turn-around delay impacts the probability of collision with carrier sensing protocols, and the power consumption when using a medium reservation preamble. The impact of the power consumption in transmit mode is less critical as transmissions are rare in ultra low power sensor networks applications. In other words, it is tolerable from an average power consumption point of view to have a relatively high transmit power (and thereby achieve a relatively long transmission range).

5.4.5 Impact of the quartz frequency tolerance

The size of the wake-up preamble is computed taking into account the quartz tolerance θ. The larger the quartz tolerance, the higher the power consumption. Fig. 5.33 shows the average power consumption and the resulting lifetime as a function of traffic intensity with a quartz tolerance of 30, 50 and 100 ppm. The largest impact is visible for inter-arrival times around 100 seconds. For $L = 100$ s, having a quartz tolerance of 100 ppm instead of 30 ppm results in a lifetime shortage of a year. In high traffic conditions, the impact is negligible. The inter-arrival being small, the wake-up preamble is small. Power consumption is dominated by data transmission. In low traffic conditions, the three curves converge as a wake-up preamble of maximum size T_W is used anyway.

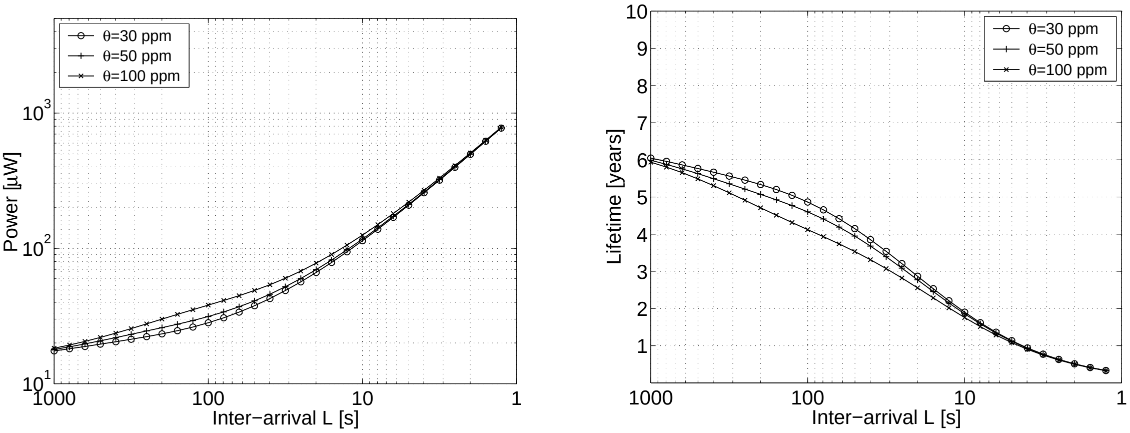

Figure 5.33: Power consumption (left) and lifetime (right) with different values for the quartz tolerance: $\theta = 30$, 50 and 100 ppm.

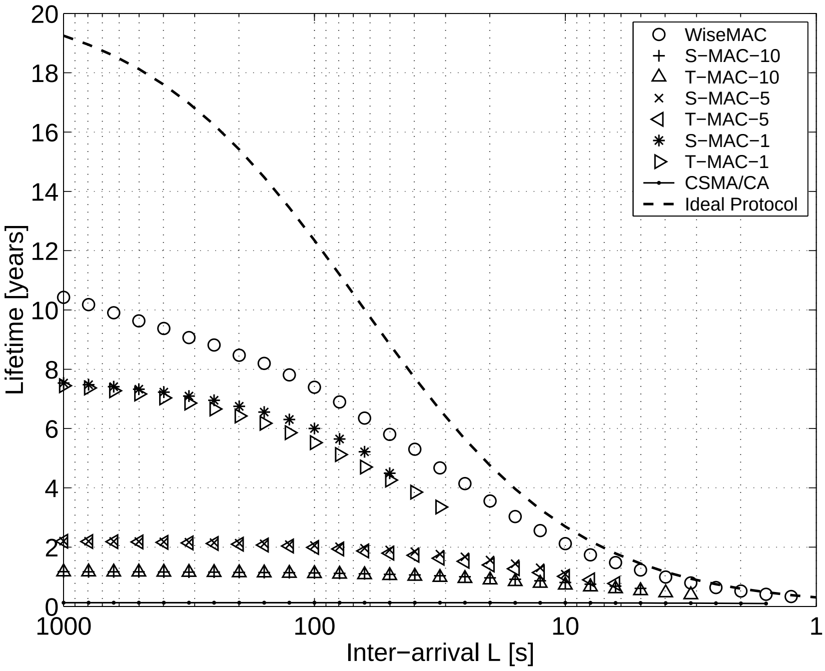

Figure 5.34: Lifetime as a function of the traffic intensity when using a single AA alkaline battery leaking 3% of its initial capacity every year.

5.4.6 Impact of the battery model

The lifetime curves have been drawn using the battery model presented in section 3.2. This model is somehow conservative as it assumes a constant leakage resulting in loosing 10% of the initial capacity every year. However, the Energizer E91 alkaline battery was recently announced to loose only 20% of its capacity within 7 years, which corresponds to 3% per year. Fig. 5.34 shows the lifetime that would be obtained when the constant leakage would be of only 3%. The impact of the lower leakage is mainly present in low traffic conditions. With an inter-arrival time of 1000 s, the lifetime is increased from 6 to 10 years.

Note that with 3% self-discharge, a battery would be empty without load after 33 years. Alkaline batteries are not designed and tested for such long operations. If such a lifetime is targeted, another battery model should be used to take more accurately aging into account.

5.5 Conclusion

WiseMAC is a single channel carrier sensing contention protocol using the preamble sampling technique to mitigate idle listening. It minimizes the length of the wake-up preamble, exploiting the knowledge of the sampling schedule of one's direct neighbors without the need for global synchronization.

WiseMAC is scalable as only local synchronization information is used. It is adaptive to the traffic load, providing an ultra low power consumption in low traffic conditions and a high energy efficiency in high traffic conditions. Thank to the "more" bit, WiseMAC can transport bursty traffic, in addition to sporadic and periodic traffic. This protocol is simple, in the sense that no complex signalling protocol is required. This simplicity can become crucial when implementing it on devices with very limited computational resources.

WiseMAC was compared to S-MAC and T-MAC both in a regular lattice topology with traffic flowing in parallel, and in a random network topology with periodic or event traffic flowing towards a sink. When forwarding packets at an interval of 1 packets every 100 seconds, the power consumption of WiseMAC was found to be 28 μW, providing 5 years of lifetime using a single AA alkaline battery. This is 75 times better than CSMA/CA and 8 times better than S-MAC and T-MAC at a duty cycle of 10%. It was shown that WiseMAC can provide simultaneously a low hop delay and a low power consumption, while S-MAC and T-MAC can only provide one or the other. Finally, it was shown that WiseMAC is able to transport a higher traffic intensity than both S-MAC-10% and T-MAC-10%.

Chapter 6

Downlink of an Infrastructure Wireless Sensor Network

6.1 Introduction

6.1.1 Problem statement

An infrastructure wireless network is composed of a number of access points interconnected through a backbone network. Each access point is serving a number of wireless sensor nodes. Such a topology, illustrated in Fig. 6.1, can be envisaged for example in smart building applications, where the Ethernet or powerline cabling can be used as a backbone network. The main characteristic of access points is that they are usually energy unconstrained. This fact can be exploited by medium access control protocols in two ways: first, an access point can listen continuously to potential uplink traffic, and secondly, an access point may send any amount of signalling traffic for free (e.g. beacons, wake-up signal). The WiseMAC protocol was designed for multi-hop networks. The work presented in this chapter was initiated to verify whether WiseMAC should also be used for the communication between an access point and a sensor node, or whether other protocols should be preferred. This work also applies to other systems where energy unconstrained nodes can play the role of base stations, such as clustered ad-hoc networks with solar powered cluster heads [69] or vehicle mounted mobile access points moving through a cloud of sensors to collect data.

In infrastructure networks, one must distinguish the downlink (from the access point to the sensor nodes) from the uplink (from the sensor nodes to the access point). For each direction, a different radio frequency and/or a different MAC protocol may be used.

In the downlink direction, the challenge is to transmit data from the access point to some sensor node, without requiring that the sensor node continuously listen to the channel. A trade-off must be made between power consumption and transmission delay.

The problem is different in the uplink direction. As the access point is not energy limited, it can listen all the time to the channel. The issue to resolve in the uplink is the multiple access to a shared medium. If the system is operated near channel capacity, this problem is very complex. However, if only moderate traffic is present on the channel, finding a energy efficient uplink MAC protocol is relatively easy. For example, the simple non-persistent CSMA protocol [58] approaches the ideal case for power conservation, with no idle listening, no overhearing and little collisions.

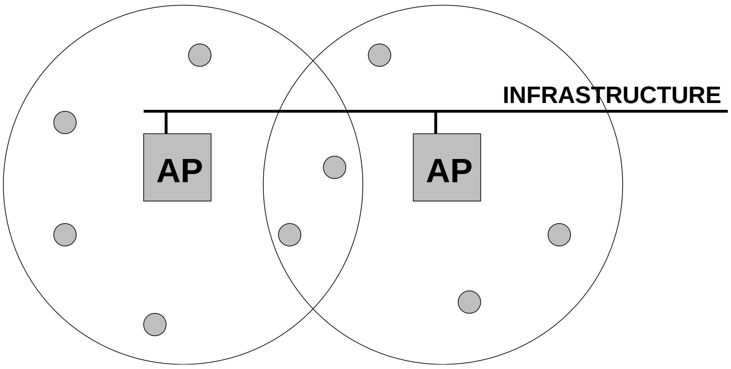

Figure 6.1: Infrastructure wireless sensor network.

In an infrastructure wireless sensor network under low traffic load, the main issue is hence the design of the downlink MAC protocol. We therefore focus on the downlink problem.

Protocols for uplink and downlink may be designed jointly. Uplink traffic can be exploited to enhance the performance of the downlink protocol. For example, an uplink packet can carry a request to transmit potentially buffered data in the downlink direction. Nevertheless, a stand-alone downlink MAC protocol is needed to guarantee a given transmission delay in periods during which uplink traffic is absent. When studying the downlink protocol, collisions with traffic belonging to some uplink protocol will not be taken into account.

Sensor networks are usually meant for the acquisition of data. Most traffic can be expected in the uplink direction. The downlink direction is foreseen to carry configuration and query requests. With such a traffic, inter-arrivals measured in minutes or hours will be common. We assume that the inter-arrival time between packets is much larger than the time needed to transmit a packet.

6.1.2 Traffic model

We consider a population of N sensor nodes under the responsibility of one access point (see Fig. 6.2). Configuration and query requests are assumed to arrive at random times. Downlink traffic will therefore be modeled following the Poisson distribution. Traffic arrives for each sensor node with an average inter-arrival L. The global downlink arrival traffic is Poisson with rate $\lambda = N/L$.

Data packets have a constant duration T_D. Control packets (pollings, acknowledgements, traffic indication map beacons) have a constant duration T_C.

We assume a low traffic where global inter-arrival $1/\lambda$ is much larger than the sum of the lengths of a data packet, of the turn-around time and of a control packet:

$$1/\lambda \gg T_D + T_T + T_C \tag{6.1}$$

6.1.3 Chapter outline

The rest of the chapter is organized as follows: Section 6.2 describes the considered protocols. The power consumption and the delay of these protocols are given in sections 6.3 and 6.4. A

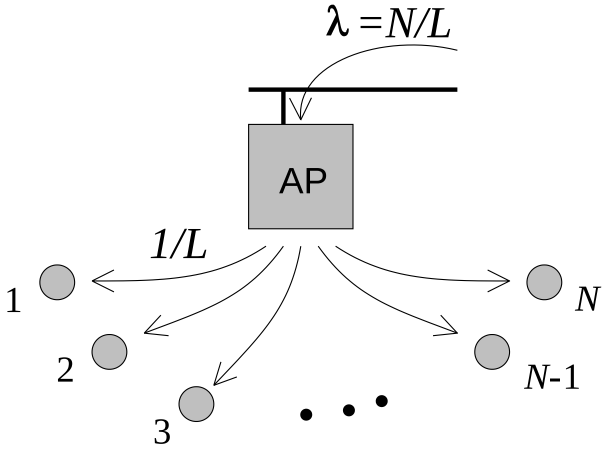

Figure 6.2: Downlink traffic model.

performance comparison is made in section 6.5. Section 6.6 contains a sensitivity analysis and section 6.7 gives conclusions.

6.2 Low power downlink MAC protocols

In this section, we describe the following four energy efficient downlink protocols:

- Ideal Protocol,
- WiseMAC,
- Periodic Terminal Initiated Polling (PTIP),
- IEEE 802.11/802.15.4 Power Save Mode (PSM)

6.2.1 Ideal protocol

In order to compare real protocols with an absolute benchmark, we define, as was done in chapter 5, an ideal protocol. Packets addressed to the sensor nodes are received by the access point from the fixed network, and forwarded to the sensor nodes. If the transmitter of the access point is free at the time of arrival, the packet is forwarded immediately on the radio channel, as shown in Fig. 6.3. If the transmitter is busy at arrival time, the packet is buffered in a FIFO queue. In the absence of traffic transmitted to them, sensor nodes are sleeping. With this ideal protocol, a sensor node magically wakes up T_S seconds before the start of the packet transmission, such that it is ready to receive the start of the packet. It then listens for the duration of the packet, sends an acknowledgement and goes back to sleep.

The purpose of this ideal protocol is to provide a target benchmark for implementable protocols. Real protocols will always consume some energy to implement the wake-up scheme. The comparison with the ideal protocol will indicate the cost of each wake-up scheme.

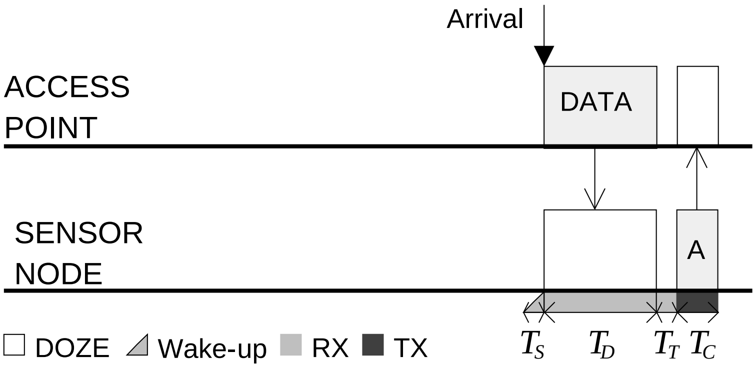

Figure 6.3: Ideal protocol.

6.2.2 WiseMAC

When using the WiseMAC protocol for the downlink of an infrastructure sensor network, the access point is the only initiator of downlink traffic. As a result, collisions are not possible. The difference in the WiseMAC protocol when used for the downlink of an infrastructure network, as compared to the multihop environment discussed in chapter 5, is that the random backoff and medium reservation collision avoidance schemes are not used. Besides this, the operation of WiseMAC is identical (see Fig. 6.4): All sensor nodes under the responsibility of an access point sample the medium with periods T_W. Their sampling offsets are independent. The base station transmits packets to sensor nodes with a wake-up preamble of length T_P centered on the expected wake-up time. The acknowledgement sent back to the base station includes an update of the synchronization information, letting the base station keep a table with the sampling schedule of all sensor nodes. The length of the wake-up preamble should cover the maximum clock drift accumulated between two transmissions. Let θ be the quartz tolerance and L be the interval between communications. As the medium reservation preamble is not used, the required length for the wake-up preamble is $T_P = T_{MR} + T_{CDC} = T_{CDC} = \min(4\theta L, T_W)$ (see section 5.2.2). In order to mitigate overhearing, long preambles are filled with copies of the data packet. The header of data packets contains a ”more” bit which indicates when more data packets addressed to the same node are waiting. When this bit is set, it indicates to the sensor node that it must continue to listen after having sent the acknowledgement. The next packet will follow (see Fig. 6.5). This scheme is particularly useful in an infrastructure network, because of the large memory resources of an access point. It is practically not limited in the number of packets that can be stored and transmitted together. This scheme allows the use of a wake-up period that is larger than the average interval between the arrivals for a given node. It permits to reduce the queuing delay at the access point, especially in the event of traffic bursts.

6.2.3 Periodic Terminal Initiated Polling - PTIP

Polling protocols are usually used to poll mobile nodes from a central access point in order to avoid collisions in the uplink direction [10]. Here, we analyze the reversed usage of polling, for the downlink direction. Such a usage of polling has not received attention from the research community, probably because it is not scalable and very inefficient in high traffic conditions. We will call this protocol PTIP, for Periodic Terminal Initiated Polling.

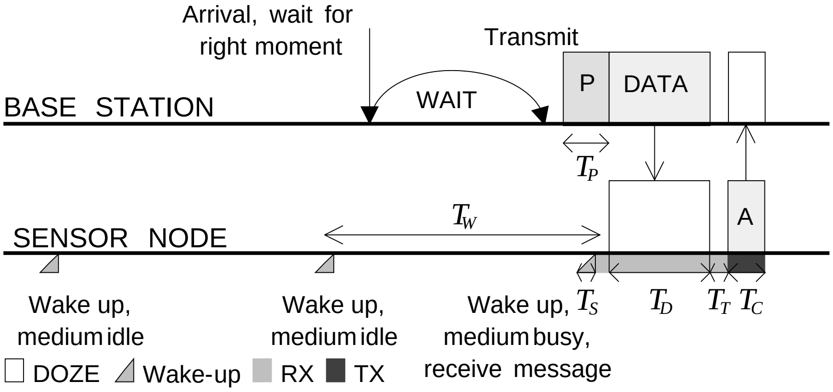

Figure 6.4: WiseMAC for the downlink of an infrastructure network.

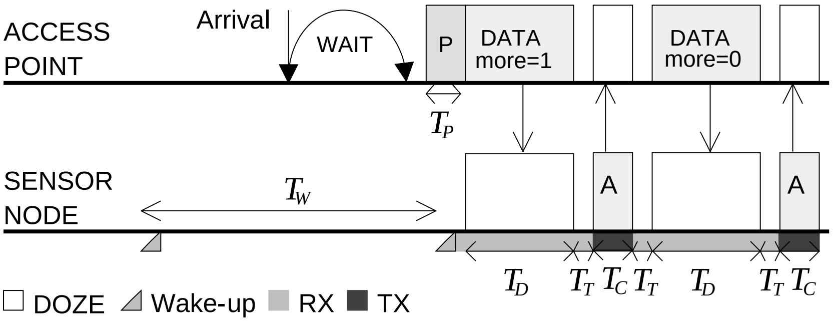

Figure 6.5: WiseMAC: Transmission of several packets using the *more* bit.

With the PTIP protocol, the access point buffers downlink traffic. Sensor nodes regularly send a poll packet to the access point to get potentially buffered data. The access point replies with a data packet if one was buffered, or with a (shorter) control packet if the queue for the requesting node was empty. To mitigate collisions between sensor nodes, poll packets are sent using the CSMA protocol. To avoid systematic contentions between synchronized nodes, the time interval between poll packet transmissions is a random variable with mean value T_W. For simplicity, we will assume in the following analysis that the polling period is constant and neglect potential systematic collisions. The principle of operation of PTIP is illustrated in Fig. 6.6. If the response to the poll packet is correctly received, the sensor node goes back to sleep until the next scheduled polling time. If the response was not received, due to a transmission error of the poll or of the response, the poll packet is retransmitted after a random backoff time. The sequence number of the last correctly received data packet must be piggy-backed on every poll packet to let the access point know when data packet transmissions must be repeated.

A *more* bit in the header of data packets indicates to sensor nodes when they must poll the access point again to download additional packets.

6.2.4 IEEE 802.11/802.15.4 Power Save Mode - PSM

A power save mode (PSM) has been specified in the IEEE 802.11 standard to allow a lower power consumption at the cost of a larger transmission delay [79]. The same scheme has been

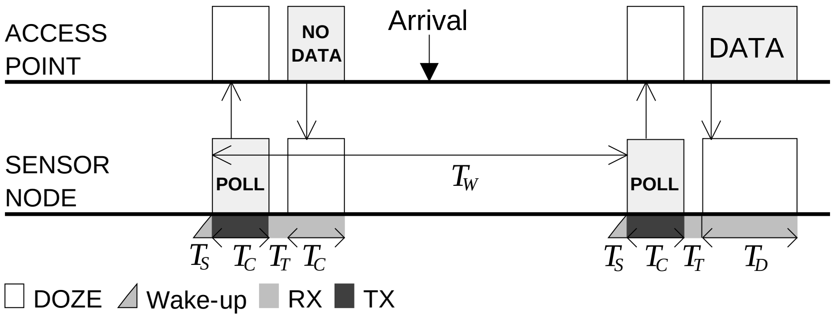

Figure 6.6: Periodic Terminal Initiated Polling (PTIP).

selected for the newer IEEE 802.15.4 standard [82]. The access point buffers incoming traffic. A beacon is periodically transmitted with period T_W. This beacon contains the *traffic indication map* (TIM), which lists the sensor nodes for which data packets have been buffered. All sensor nodes wake up regularly to receive the TIM. If a sensor node finds its address in the TIM, its sends a poll packet to the access point, demanding the transmission of the announced data packet.

The standard requires the access point to reply to a poll packet after a given delay (10 μs in DSSS IEEE 802.11b). In practice, it is difficult for the access point software to find the right packet and prepare it for transmission within the specified delay. Instead, the access point replies to the poll packet with an acknowledgement packet. This instructs the sensor node to remain in listening mode. As soon as possible, the access point sends the data packet, which is then acknowledged back by the sensor node. In summary, the polling procedure is composed of four packet transmissions: POLL-ACK-DATA-ACK. We are interested here in the basic performance of protocols that would use a traffic indication map. For a fair comparison with the other protocols, we consider a version of the PSM protocol that is fully optimized for low power operation. We assume first that an access point replies to a poll packet with a data packet and secondly that a data packet is not acknowledged (as in PTIP, the acknowledgement is piggy-backed on the following poll packet). This procedure is illustrated in Fig. 6.7.

A *more* bit in the header of data packets indicates to sensor nodes when they must poll the access point again to download the additional packets.

In Fig. 6.7, the periods during which a sensor node is receiving are marked with a light gray bar below the time line. The periods during which a sensor node is transmitting are marked with a dark gray bar. One can observe that the sensor node starts listening T_{Sync} seconds before the actual start of the TIM packet. This behavior is required to compensate for the clock drift between the quartz running on the access point and on a sensor node. It was shown in the equivalent case of TDMA (chapter 4, section 4.2.2), that the duration of the required synchronization period is $T_{Sync} = 4\theta T_W$ in the worst case. Taking the average over the possible quartz inaccuracies, assuming a uniform distribution of the quartz inaccuracies, we have

$$\overline{T}_{Sync} = 2\theta T_W \tag{6.2}$$

A sensor node may learn the relative difference between the frequency of its clock and of the clock running on the access point, and reduce the required synchronization time T_{Sync}.

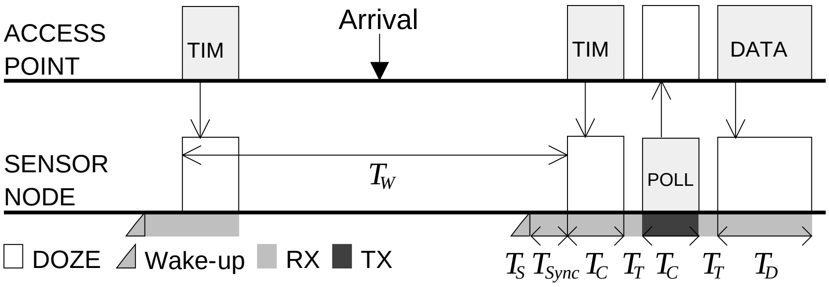

Figure 6.7: Optimized Power Save Mode (PSM).

This optimization has not been considered here. It will be seen later, that even without this optimization, the average power required for periodic re-synchronization is very small.

6.2.5 Adaptability of the wake-up period

Different nodes may have different delay requirements. For example, a light switch must react to a command within less than a second, while a meter reader could answer to an automated data collecting center within minutes. It is important for a MAC protocol to allow different nodes to use different wake-up periods within the same network. This feature is available with all three protocols:

- With WiseMAC, the access point has to remember the sampling period of each sensor node, and compute the wake-up preamble length accordingly,
- With PTIP, each node is free to choose its polling period. The access point doesn't even have to know them,
- With PSM, a node may decide to listen to each TIM broadcast. It may also choose to skip some of them. The data frames will remain buffered as long as they have not been polled. The wake-up period can hence be freely chosen as a multiple of the TIM broadcast period.

6.3 Power consumption

In this section, we derive analytical expressions for the power consumption of the four protocols: Ideal, WiseMAC, PTIP and PSM. This analysis brings an insight into the sources of energy waste of these protocols, and allows their comparison.

6.3.1 Power consumption of the ideal protocol

We start the analysis with the easiest case: the ideal protocol. Packets received from the fixed network are queued and transmitted in FIFO order on the radio channel. Sensor nodes "magically" wake up just at the right time to receive the packets addressed to them. Packets are acknowledged by the sensor nodes.

A sensor node receives one packet of length T_D every L seconds on average. The energetic increment cost of receiving one packet is $E_{IDEAL} = \widehat{P}_S T_S + \widehat{P}_R(T_D + T_T) + \widehat{P}_T T_C$. This is the increment compared to what would have been consumed when being in the doze state. In this sum, $\widehat{P}_S T_S$ is the energy required to power-on the transceiver into receive mode. $\widehat{P}_R T_D$ is the energy to receive the packet. $\widehat{P}_R T_T$ is the energy to turn-around the transceiver into transmit mode and $\widehat{P}_T T_C$ is the energy to send the acknowledgement. On average, we obtain a power consumption of

$$P_{IDEAL} = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R (T_D + T_T) + \widehat{P}_T T_C}{L} \tag{6.3}$$

In this expression, it is important not to neglect the setup and turn-around phases, as the energy consumed in these states can exceed the energy consumed to receive the data packet. Indeed, with a high transmission rate, the energy $\widehat{P}_R T_D$ needed to receive a packet of a few tens of bytes may be smaller than the setup energy $\widehat{P}_S T_S$.

6.3.2 Power consumption of WiseMAC

The power consumption of a sensor node using the WiseMAC protocol and receiving packets with an average inter-arrival L is given by

$$\begin{aligned} P_{WiseMAC} = P_Z &+ \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} \\ &+ \frac{\widehat{P}_R (\overline{T}_{LP} + T_D + T_T) + \widehat{P}_T T_C}{L} \\ &+ \widehat{P}_R (N-1) \frac{\overline{T}_O}{L} \end{aligned} \tag{6.4}$$

where the first term accounts for the power consumption in doze state, the second term accounts for the increment caused by the sampling activity, the third term the increment caused by the reception and acknowledgement, and the last term the overhearing of the transmission to the $N-1$ other nodes. $\overline{T}_{LP}$ is the average duration during which a sensor node that is the destination of a message listens to the wake-up preamble. $\overline{T}_O$ is the average duration during which a node overhearers a transmission. As introduced in chapter 5, expressions (5.21) and (5.22), we have

$$\overline{T}_{LP} = 2\theta L \left(1 - e^{-\frac{T_D}{4\theta L}}\right)$$

and

$$\overline{T}_O = \frac{T_D^2 + 12 T_D \theta L}{2 T_W} \left(1 - e^{-\frac{T_W}{4\theta L}}\right)$$

Expression (6.4) is very similar to the power consumption in the multi-hop case (5.19). The difference is that transmission power consumption is not taken into account.

6.3.3 Power consumption of PTIP

Assuming no collisions between poll packets, the average power consumed by PTIP to receive data packets with an average inter-arrival L is given by

$$\begin{aligned} P_{PTIP} = P_Z + e^{-\frac{T_W}{L}} \frac{\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R (T_T + T_C)}{T_W} \\ + \frac{\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R (T_T + T_D)}{L} \end{aligned} \tag{6.5}$$

The fist term is the power consumed in doze mode. The second term accounts for useless polling, i.e. polling done when no data packet was waiting. The third term accounts for the polling of buffered data packets.

To derive expression (6.5), let us first compute the energy spent in a period T_W. With PTIP, as said previously, a sensor node sends a poll packet every T_W seconds to the access point. Data packets addressed to this sensor node arrive at the access point following a Poisson process of rate $1/L$. With probability $e^{-\frac{1}{L}}$, no data packet will be waiting for this node. The access point will reply with a short control packet. This event will consume an energy $\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R T_T + \widehat{P}_R T_C$ (setup, send the poll, turn around, receive the control packet). With probability $1 - e^{-\frac{1}{L}}$, $k \geq 1$ data packets will be waiting in the buffer. In cases where $k > 1$ data packets are buffered, they are all downloaded in a row thanks to the *more* bit which indicates when further data packets are waiting. Let K_1 be the average number of buffered data packets, given that at least one data packet is present. The sensor node will consume an energy $K_1 \left(\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R (T_T + T_D) \right)$ to download all data packets[1]. The average energy consumed by PTIP to receive the data packets that have been buffered at the access point in a period T_W is hence given by

$$\begin{aligned} E_{PTIP} = e^{-\frac{T_W}{L}} \left(\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R (T_T + T_C) \right) \\ + (1 - e^{-\frac{T_W}{L}}) K_1 \left(\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R (T_T + T_D) \right) \end{aligned} \tag{6.6}$$

K_1 can be computed as follows: With a Poisson arrival of rate $1/L$, the average number of arrivals in a period T_W is $K = \frac{T_W}{L}$. Here, we are interested in the average number of arrivals, given that we have at least one arrival. There will be zero arrivals with probability $e^{-\frac{T_W}{L}}$ and at least one arrival with probability $1 - e^{-\frac{T_W}{L}}$. We know that $K = e^{-\frac{T_W}{L}} \cdot 0 + \left(1 - e^{-\frac{T_W}{L}} \right) \cdot K_1$. Hence,

$$K_1 = \frac{\frac{T_W}{L}}{1 - e^{-\frac{T_W}{L}}} \tag{6.7}$$

Combining (6.7) with (6.6), dividing by T_W, adding P_Z, one obtains (6.5).

In this analysis, we have neglected the power consumption of retransmissions. As poll packets are sent using the CSMA protocol, there is a risk to see poll packets sent by different sensor nodes collide. Expression (6.5) is hence accurate only for large polling periods and low downlink traffic conditions. This point will be addressed in more details when looking at the transmission delay.

---

[1] If several data packets are downloaded in a row, the transceiver will need an energy $P_S T_S$ to turn the transceiver on before sending the first poll packet. A smaller energy $P_R T_T$ will be needed before to send the subsequent poll packets. To be accurate, one must hence consider that expression (6.5) is an upper bound on the power consumption of PTIP.

6.3.4 Power consumption of PSM

Under the assumptions that a poll packet is directly answered with a data packet, and that a data packet is acknowledged in the subsequent poll, the average power consumed by PSM to receive packets with an average inter-arrival L is given by

$$P_{PSM} = P_Z + 2\theta \widehat{P}_R + \frac{\widehat{P}_S T_S + \widehat{P}_R T_C}{T_W} + \frac{\widehat{P}_T T_C + \widehat{P}_R (T_D + 2T_T)}{L} \tag{6.8}$$

The first term represents the power consumed in doze mode. The second term, $2\theta\widehat{P}_R$, accounts for the time spent listening to the channel to cover the drift between the access point clock and the sensor node clock. The third term represents the power consumed to power-on and listen to the beacon of length T_C every T_W seconds. Finally, the fourth term accounts for the reception and acknowledgement of data packets.

To derive expression (6.8), let us first compute the energy spent in a period T_W. We assume that all traffic arriving during a period T_W is polled during the following period T_W. We do not consider bursty arrivals that require several periods to be downloaded. Multiple arrivals for a single sensor node are detected by the sensor node thanks to the *more* bit. Under these assumptions, the energy spent by a sensor node in a period T_W is composed of the energy to receive the beacon $\widehat{P}_S T_S + \widehat{P}_R(\overline{T}_{Sync} + T_C)$ and the energy to download all $\frac{T_W}{L}$ buffered packets. We have

$$E_{PSM} = \widehat{P}_S T_S + \widehat{P}_R(\overline{T}_{Sync} + T_C) + \frac{T_W}{L}\left(\widehat{P}_T T_C + \widehat{P}_R(T_D + 2T_T)\right) \tag{6.9}$$

$\widehat{P}_S T_S + \widehat{P}_R(\overline{T}_{Sync} + T_C)$ represents the energy to power-on the transceiver, to listen to the medium until the beacon transmission starts and to listen to the beacon. $\widehat{P}_T T_C + \widehat{P}_R(T_D + 2T_T)$ represents the energy to send the poll packet and receive the data. Here, we count also the turn-around phases before sending the poll and before receiving the data. Combining (6.9) with (6.2), dividing by T_W, adding P_Z, one obtains (6.8).

As in the case of PTIP, we have neglected the impact of retransmissions of poll packets required in case of collisions. This point will also be addressed in more details when looking at the transmission delay.

6.4 Transmission delay

The transmission delay is defined as the time elapsed between the arrival of a packet at the access point and the end of its reception by the destination sensor node. In this section, using an approximation analysis, we derive the transmission delay obtained using the four protocols under investigation.

6.4.1 Delay with the ideal protocol

In the case of the ideal protocol, packets of fixed length T_D arrive following a Poisson process of rate λ. They are buffered in a FIFO buffer and transmitted sequentially. The average delay of this M/D/1 queue is given by

$$D_{IDEAL} = T_D + \frac{\lambda m^2}{2(1-\lambda m)}$$

where $m = T_D + T_T + T_C + T_T$ (see [8]). As $1/\lambda \gg T_D + T_T + T_C$ (low traffic assumption (6.1)), we have

$$D_{IDEAL} \approx T_D \tag{6.10}$$

6.4.2 Delay with WiseMAC

To compute the average delay with WiseMAC, we will consider that the access point has one FIFO queue for each sensor node. Incoming packets are stored in the FIFO queue of the respective destination. Following some policy, the access point chooses a queue to serve. Thanks to the *more* bit, all packets are sent to the chosen sensor node in a row, starting at the instant when this sensor node samples the medium. To avoid having a large number of packets in the queues, one must choose a policy that maximizes the channel utilization. Once a queue has been served, such a policy is to choose, as the next queue to be served, the one of the sensor node which is next going to sample the medium. Based on this policy, and as $1/\lambda \gg T_D + T_T + T_C$ (low traffic assumption (6.1)), we can assume that every queue is served after at most T_W seconds.

We will start by computing the transmission delay considering that a single packet is sent towards a given sensor node, i.e. only one packet destined to this node has arrived at the base station in the period of duration T_W preceding the transmission. We will next see that the average time needed to transmit possible additional packets is negligible.

A data packet is transmitted with a preamble of length $T_P = \min(4\theta l, T_W)$. Depending on the value of the interval l between two communications with a given node, the three following scenarios are possible (see also Fig. 5.11):

A) $T_P < T_D$,

B) $T_W > T_P \geq T_D$,

C) $T_P = T_W$.

In scenario A, the packet must first wait for the right transmission time. The duration of the wait time can vary between 0 (if this packet is lucky) and T_W. On average, the wait time is equal to $\frac{T_W}{2}$. Then, one must count the time to transmit the wake-up preamble and the data. We have $D_A = \frac{T_W}{2} + 4\theta l + T_D$.

Scenario B is different from scenario A because the wake-up preamble is composed of copies of the data frame. The first delay is again the waiting delay $\frac{T_W}{2}$. During the transmission of the wake-up preamble, the destination node may wake up at any instant. On average, it will wake up after $\frac{T_P}{2} = 2\theta l$ seconds. Once awake, the node waits for the start of the next copy of the data packet ($\frac{T_D}{2}$ on average) and then receives the data packet. We have $D_B = \frac{T_W}{2} + 2\theta l + \frac{T_D}{2} + T_D$.

In scenario C, the synchronization is assumed to be lost. The wake-up preamble has the length of the sampling period. The transmission is started as soon as the packet is received by the base station. The destination will, on average, sample the medium after $\frac{T_W}{2}$ seconds of preamble transmission. It will then wait for the start of the next copy of the data packet ($\frac{T_D}{2}$ on average) and then receive the data packet. We have $D_C = \frac{T_W}{2} + \frac{T_D}{2} + T_D$.

The average value of the delay can be computed as $\overline{D} = \int_0^{\frac{T_D}{4\theta}} D_A \frac{1}{L} e^{-\frac{l}{L}} dl + \int_{\frac{T_D}{4\theta}}^{\frac{T_W}{4\theta}} D_B \frac{1}{L} e^{-\frac{l}{L}} dl + \int_{\frac{T_W}{4\theta}}^{\infty} D_C \frac{1}{L} e^{-\frac{l}{L}} dl$, which gives

$$D_{WiseMAC} = T_D + \frac{T_W}{2}\left(1 - e^{-\frac{T_W}{4\theta L}}\right) + 2\theta L\left(2 - e^{-\frac{T_D}{4\theta L}} - e^{-\frac{T_W}{4\theta L}}\right) \tag{6.11}$$

Let us now consider the case with multiple packets in the queue for a given sensor node. As previously defined in (6.7), K_1 is the average number of packets in the queue, given that at least one packet is present. The duration needed to transmit K_1 packets in a row is given by $T_D + (K_1 - 1)(T_T + T_C + T_T + T_D)$ (see Fig. 6.5). With T_W smaller than L, having more than one packet becomes a rare event. We have $(K_1 - 1) \approx 0$. With T_W larger than L, we have $K_1 \approx \frac{T_W}{L}$. Using the low traffic assumption $L > 1/\lambda \gg T_D + T_T + T_C$, one can write $T_D + (\frac{T_W}{L} - 1)(T_T + T_C + T_T + T_D) < \frac{T_W}{L}(T_T + T_C + T_T + T_D) \ll T_W$. When T_W is large, the time needed to transmit all K_1 packets is hence negligible in comparison with the average waiting time $T_W/2$. Expression (6.11) is hence also valid when multiple packets are transmitted in a row.

6.4.3 Delay with PTIP

With PTIP, the delay is first composed of the time between the arrival of a packet and the time the next polling packet is received from the destination of this packet. As a sensor node sends a poll packet on average every T_W seconds, the average is $\frac{T_W}{2}$. The second component is the time needed to send the packet of interest ($T_T + T_D$, see Fig. 6.6). Here again, more than one packet may have been buffered for the same sensor node. Using the same reasoning as presented for WiseMAC, one can show that the average additional time needed to transmit the following packets is either much smaller than T_D for small T_W or much smaller than $\frac{T_W}{2}$ for large T_W. We have

$$D_{PTIP} \approx \frac{T_W}{2} + T_T + T_D \tag{6.12}$$

Here, we assume that no collision occurs between the poll packets sent by different sensor nodes. The download periods of different sensor nodes shall hence very rarely overlap. For this to be true, we need to set a condition on the total traffic present on the channel. The bandwidth used by the data traffic and by the poll packets regularly sent by the N sensor nodes must be far from the channel capacity: $\lambda(T_C + T_T + T_D) + \frac{T_C}{T_W}N \ll 1$. We know from (6.1) that $\lambda(T_C + T_T + T_D) \ll 1$. For expression (6.12) to be accurate, we therefore need to make the following assumption:

$$T_W \gg N T_C \tag{6.13}$$

In section 6.5, we evaluate expressions (6.5) and (6.12) for T_W in the interval $[\beta N T_C, \infty]$, where β is a relatively large number. $\beta = 100$ has been chosen to ensure the usage of less that

1% of the capacity by poll packets.

6.4.4 Delay with PSM

With PSM, packets arriving at the access point before a beacon transmission are polled by the destination sensor node after the beacon transmission. As $1/\lambda \gg T_C + T_T + T_D$ (low traffic assumption (6.1)), we can assume that all packets received in a period T_W are transmitted in the following period T_W.

As illustrated in Fig. 6.7, the delay is composed of three components: The first component is the time between the arrival of a packet and the start of the transmission of the TIM beacon. As a given packet may arrive at any time with equal probability, the average will be $\frac{T_W}{2}$. The second component is simply T_C, the time needed to broadcast the TIM. The third component is the time needed for the packet of interest to be polled.

If the period T_W is large compared to the inter-arrival $1/\lambda$, there will be a number of sensor nodes entering in contention after the broadcast of the TIM. Using non-persistent CSMA, the time needed to resolve the collision is not bounded. If the traffic is small, the average duration of the collision resolution interval can however be expected to be smaller than the period T_W. The computation of the duration of this collision resolution interval with non-persistent CSMA is unfortunately a problem for which, to our knowledge, no analytical solution is available. A formula for the steady state transmission delay using non-persistent CSMA has been given by Kleinrock and Tobagi in [58]. In our case, traffic is composed of a single burst, making results for steady state unsuited. Using more complex collision resolution protocols, such as the tree algorithms proposed by Capetanakis [9], one can derive an average duration of the collision resolution interval. We do not consider such results for the sake of simplicity and because, as will be shown, we do not need to consider collisions to compare the protocols in a useful manner. To avoid the problem of potential collisions, we consider with PSM a traffic sufficiently small such that at most one packet is received on average in a period T_W. In the case of PSM, we assume

$$T_W \leq 1/\lambda \tag{6.14}$$

Under this assumption, the delay becomes simply

$$D_{PSM} \approx \frac{T_W}{2} + 2T_C + 2T_T + T_D \tag{6.15}$$

The TIM broadcast period should clearly be chosen larger than $2T_C + 2T_T + T_D$ to give enough time for the polling of at least one packet. Expressions (6.8) and (6.15) shall hence be evaluated for $T_W \in [2T_C + 2T_T + T_D; 1/\lambda]$.

6.5 Performance comparison

In this section, we compare the power consumption and delay performances of the WiseMAC, PTIP and PSM protocols as a function of the protocol parameter T_W. The choice of T_W permits to make a trade-off between the transmission delay and the power consumption.

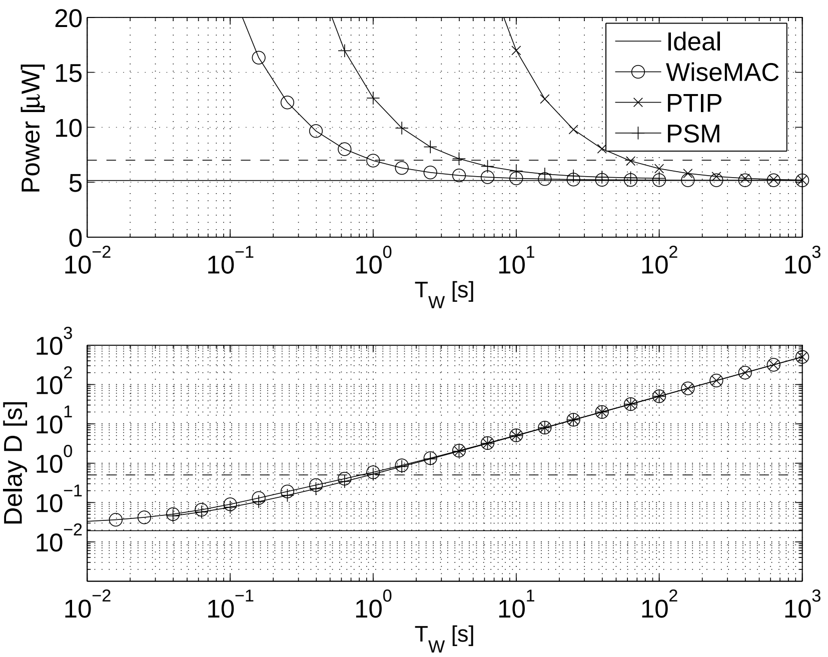

Figure 6.8: Power consumption and delay of WiseMAC, PTIP and PSM as a function of the wake-up period T_W ($L = 1000$ s).

The performance of the protocols is also influenced by a number of system parameters. For the radio transceiver parameters, we will consider those of the WiseNET transceiver, introduced in section 3.3.

The remaining parameters have been chosen as follows: The frequency tolerance of the quartz is chosen to be $\theta = \pm 30$ ppm. The length of the data and acknowledgement packets is chosen to be respectively 60 and 10 bytes, as in the preceding chapter. Assuming that the bit/frame synchronization and frame checking overhead amounts to 6 bytes (2 bytes for bit synchronization, 2 bytes start frame delimiter and 2 bytes frame check sequence), this would leave 6 bytes for useful signalling data. At 25 kbps, this yields packet durations of $T_D = 19.2$ ms and $T_C = 3.5$ ms. We consider $N = 10$ sensor nodes and an inter-arrival per sensor node of $L = 1000$ s = 16.6 min. Remember that this traffic is supposed to consist of configuration and query commands sent by the sensor network controller. Such large inter-arrivals make hence sense in this context. The sensitivity of the results to variations of the system parameters will be discussed in section 6.6.

Fig. 6.8 shows the power consumption P and the delay D as a function of T_W. The horizontal line in the upper plot represents the power consumption of the ideal protocol. In this case, we have $P_{IDEAL} = P_Z + (\widehat{P}_S T_S + \widehat{P}_R(T_D + T_T) + \widehat{P}_T T_C)/L = 5.09$ μW. With $L = 1000$, the incremental cost due to the data reception is only 0.09 μW, a small value compared to $P_Z = 5$ μW.

In the lower plot, the horizontal line represents the minimum delay that would be obtained with the ideal protocol, i.e. $D_{IDEAL} = T_D = 19.2$ ms. WiseMAC and PSM approach this limit for small values of T_W, but at a high power cost. For large values of T_W, the three curves converge to $D = T_W/2$, because the transmission delay becomes negligible compared to the waiting delay.

Using both plots, one can choose a protocol and a value for the parameter T_W, making a trade-

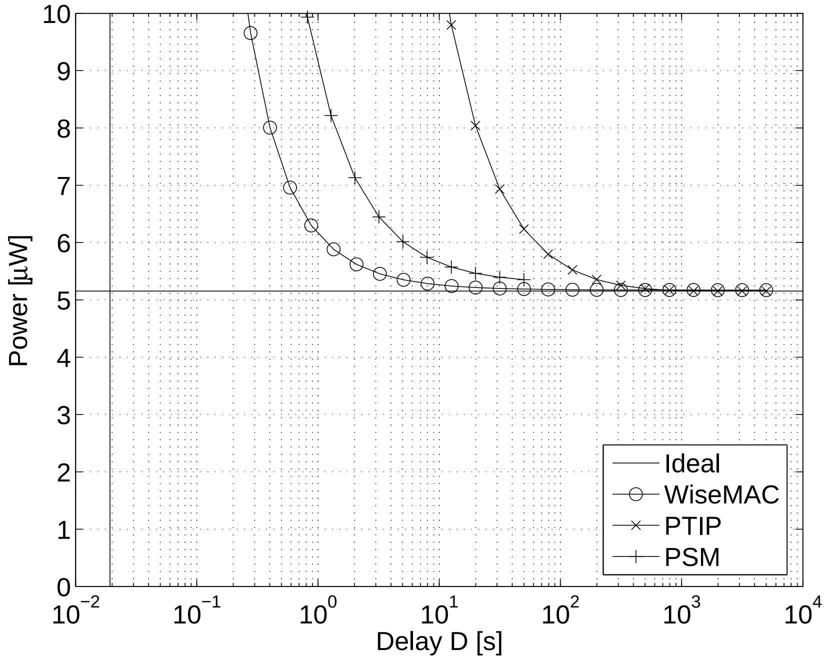

Figure 6.9: Power-delay characteristics of WiseMAC, PTIP and PSM ($L = 1000$ s).

off between the consumed power and the average transmission delay. To compare the protocols, one can combine both plots and draw the power-delay characteristic for a varying T_W, as shown in Fig. 6.9. On this graph, the horizontal line represents the ideal power consumption and the vertical line the ideal delay. The power consumption of PSM is drawn only up to a delay of 50 s. For larger values of the delay, assumption (6.14) would be violated.

One can see that WiseMAC consumes less power than PTIP and PSM. The cost of receiving data being negligible when $L = 1000$ s, this can be understood by comparing the basic cost of their respective wake-up scheme. With an infinitely low traffic ($L \to \infty$), the power consumption of the 3 protocols becomes

$$\lim_{L \to \infty} P_{WiseMAC} = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_R T_I}{T_W} \tag{6.16}$$

$$\lim_{L \to \infty} P_{PTIP} = P_Z + \frac{\widehat{P}_S T_S + \widehat{P}_T T_C + \widehat{P}_R (T_T + T_C)}{T_W} \tag{6.17}$$

$$\lim_{L \to \infty} P_{PSM} = P_Z + 2\theta \widehat{P}_R + \frac{\widehat{P}_S T_S + \widehat{P}_R T_C}{T_W} \tag{6.18}$$

With WiseMAC, the transceiver powers-on every T_W to listen to the channel during T_I, the duration of a radio symbol. With PTIP, the transceiver periodically sends a poll packet and receives a reply. With PSM, the transceiver periodically receives a TIM packet. As the duration of a TIM packet is always larger than the duration of a modulation symbol, the wake-up scheme of WiseMAC consumes less than the one of PSM. As receiving a TIM packet consumes less than transmitting a poll packet and receiving a reply, PSM consumes less than PTIP.

With delays above 300 s, WiseMAC and PTIP converge to the ideal power consumption, which is about P_Z when $L = 1000$. PSM converges to a value that is $2\theta\widehat{P}_R$ larger[2].

Let us assume that, based on the requirements of some application, one would choose to have an average delay of 0.5 s (and a maximum delay of 1 s). As can be seen in Fig. 6.8, where the 0.5 s average delay is represented by an horizontal dashed line, this would imply to select $T_W = 1$ s. Receiving one data packet every $L = 1000$ s, the power consumption of WiseMAC would amount to 7 μW, only 2 μW above the doze power consumption. When powered by a single AA alkaline battery of 2.6 Ah capacity with a constant power leakage of 27 μW, this power consumption would translate into a lifetime of 8 years (without uplink traffic). See section 3.2 for a description of the battery model. For the same delay, the power consumption of PSM amounts then to 12 μW, almost two times more. To consume only 7 μW with PSM, the average delay must be extended to 2 seconds, i.e. four times more.

With the PTIP protocol, such a low delay cannot be reached as the required wake-up period would cause too many collisions (assumption (6.13)). If an average delay of about 30 seconds can be accepted, choosing $T_W = 60$ s would give with PTIP a power consumption in the order of 7 μW. PTIP is of interest for applications that can tolerate a large delay. Its advantage compared to WiseMAC and PSM lies in its extreme simplicity to implement. The PTIP protocol can hence become an excellent choice for cost sensitive and delay tolerant applications. It must also be noticed that uplink traffic can be used to piggy-back poll requests. If an application requires periodic uplink transmission, then the PTIP protocol can be implemented for the downlink at no cost.

6.6 Sensitivity analysis

6.6.1 Sensitivity to traffic

In section 6.5, we have considered a constant average traffic with $L = 1000$ s. We have selected a value $T_W = 1$ s for the wake-up period, as a trade-off between the power consumption and the delay. In this section, we keep $T_W = 1$ s constant and vary the traffic L. Fig. 6.10 shows the power consumption of the Ideal, WiseMAC and PSM protocols[3].

If the traffic increases (L decreases), the power consumption increases gracefully for both WiseMAC and PSM. Their power consumptions approach what would be obtained with the ideal protocol.

If the traffic decreases (L increases), the power consumption decreases gracefully for both WiseMAC and PSM. The power consumption of PSM converges to expression (6.18), shown as the upper horizontal dashed line in Fig. 6.10. The power consumption of WiseMAC converges to expression (6.16), shown as the lower horizontal dashed line in Fig. 6.10. Both protocols support gracefully an increasing inter-arrival between data packets. We will see in section 6.6.3 that this property is brought to WiseMAC by the repetition of the data frame in the preamble.

6.6.2 Scalability

We have seen that the overhearing component in the power consumption of WiseMAC (6.4) is proportional to the number of potential overhearers. We are now interested in measuring the

---

[2] With $T_W > 1/\lambda$, the delay with PSM may increase because of collisions between the poll packets of several sensor nodes following the TIM transmission. This behavior does not appear because of the simplifications made.

[3] The PTIP protocol is not shown because assumption (6.13) is not valid with $T_W = 1$ s. In any case, the power consumption of PTIP would be above 20 μW when $T_W = 1$ s.

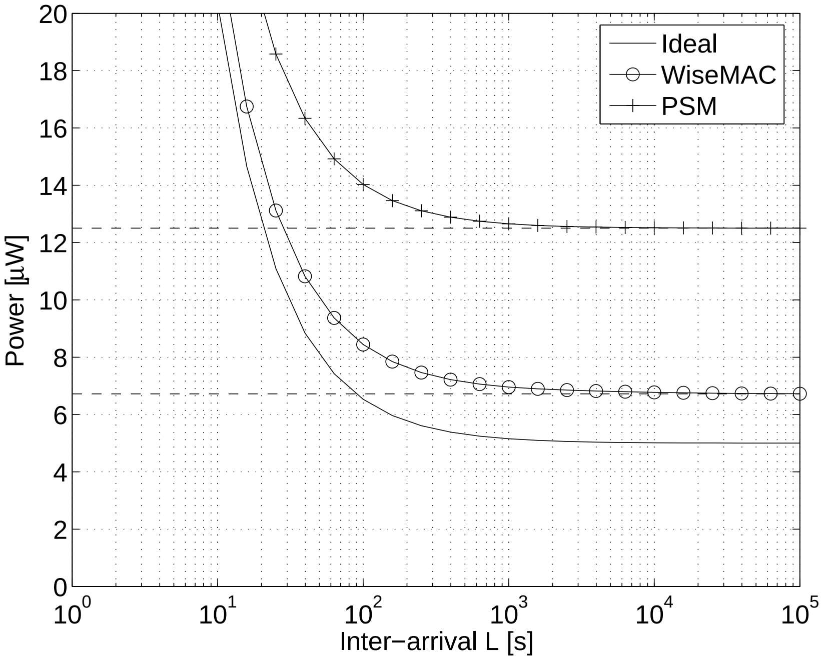

Figure 6.10: Power consumption as a function of the inter-arrival L ($T_W = 1$ s).

scalability of WiseMAC. Fig. 6.11 shows the power consumption of the WiseMAC and PSM protocols as a function of the number of sensor nodes. The traffic for each sensor node remains constant. Increasing the number of sensor nodes therefore increases the global traffic. This shows the impact of adding more nodes to the network. The power consumption is plotted for different values of the per-node inter-arrival L.

One can observe in expression (6.8) that the power consumption of PSM is independent of N. As long as $L/N \geq T_W$ (assumption (6.14)), the PSM protocol remains perfectly scalable[4]. This is made possible through the combined use of the polling technique (which avoids overhearing) and the regular broadcast of the traffic indication map (which avoids useless polling traffic). The PSM curve is stopped when $L/N = T_W$. Above this value, potential collisions between poll packets may increase the power consumption of PSM and thereby degrade its scalability.

It can be seen in Fig. 6.11 that although WiseMAC includes an overhearing component, it does scale well. It remains better than the corresponding curve for PSM up to thousands of sensor nodes. With a small inter-arrival ($L = 100$ s), the wake-up preamble is small compared to the sampling period, and overhearing is mitigated in a probabilistic way ($T_P + T_D \ll T_W$). With a large inter-arrival ($L = 100000$ s), the wake-up preamble is as long as the sampling period. Overhearing is mitigated through the repetition of the data frame in the wake-up preamble. Overhearers listen on average during $T_D/2 + T_D$ seconds before discovering that the packet is not addressed to them, and go back to sleep.

Finally, we can mention that the PTIP protocol (not shown in Fig. 6.11) is clearly not scalable

---

[4] The scalability is only perfect with our model of the protocol, where the size T_C of the traffic indication map is constant. In a real implementation, with an increasing number of sensor nodes for which a constant traffic is addressed, the size of the traffic indication map will have to grow. Bitmap coding of the destination addresses may need to be replaced with a list of full addresses. Techniques used in paging systems may be of interest to find an energy efficient solution [70].

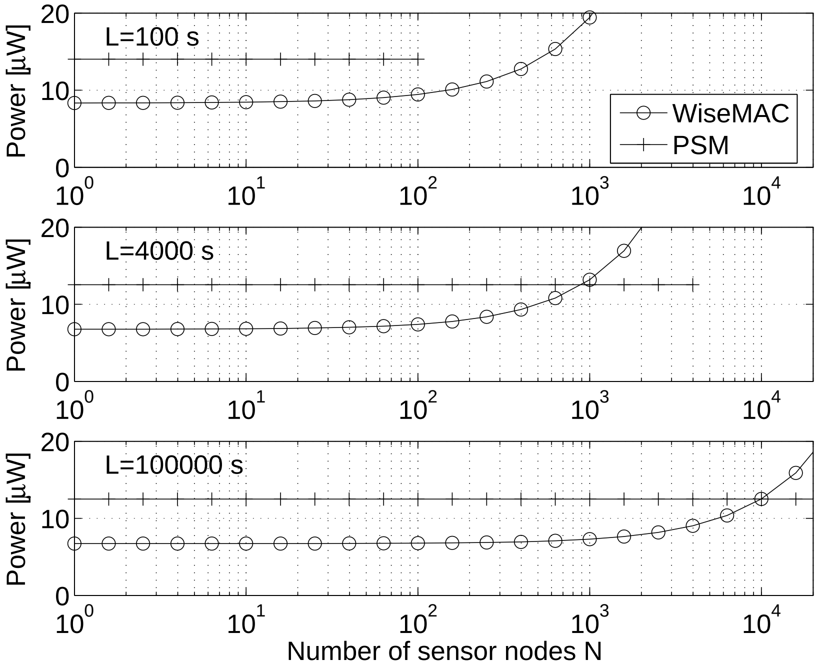

Figure 6.11: Power consumption as a function of the number of sensor nodes N, for different values of L ($T_W = 1$ s).

because of the potential collisions between the poll packets regularly sent by the sensor nodes.

6.6.3 Impact of the data frame repetition in the WiseMAC preamble

Without the repetition of the data frame in the wake-up preamble, WiseMAC may present a large overhearing overhead, depending on the value of the inter-arrival. As overhearing is proportional to the number of nodes, this would degrade the scalability of WiseMAC.

Let us call WiseMAC* the simpler version of the protocol, where the preamble is composed of alternating bits. The power consumption of WiseMAC* is given by

$$\begin{aligned} P_{WiseMAC^*} = P_Z &+ \frac{\widehat{P}_S T_S + \widehat{P}_I T_I}{T_W} \\ &+ \frac{\widehat{P}_R(\overline{T}^*_{LP} + T_D + T_T) + \widehat{P}_T T_C}{L} \\ &+ \widehat{P}_R (N-1) \frac{\overline{T}^*_O}{L} \end{aligned} \tag{6.19}$$

$\overline{T}^*_{LP}$ can be computed similarly as $\overline{T}_{LP}$ (expression (5.21)). Assuming that the clock drifts of both the access point and the destination sensor nodes are uniformly distributed in $[-\theta; +\theta]$, the destination sensor node listens on average to the wake-up preamble during $T_P/2$. With a periodic traffic with period L, we would have an average listening duration of $\min(2\theta L, T_W/2)$. Let L be an exponentially distributed random variable, and l be a given value that it takes. With Poisson traffic, we hence have $\overline{T}^*_{LP} = \int_0^{T_W/4\theta} 2\theta l \frac{1}{L} e^{-\frac{l}{L}} dl + \int_{T_W/4\theta}^{\infty} \frac{T_W}{2} \frac{1}{L} e^{-\frac{l}{L}} dl$, giving:

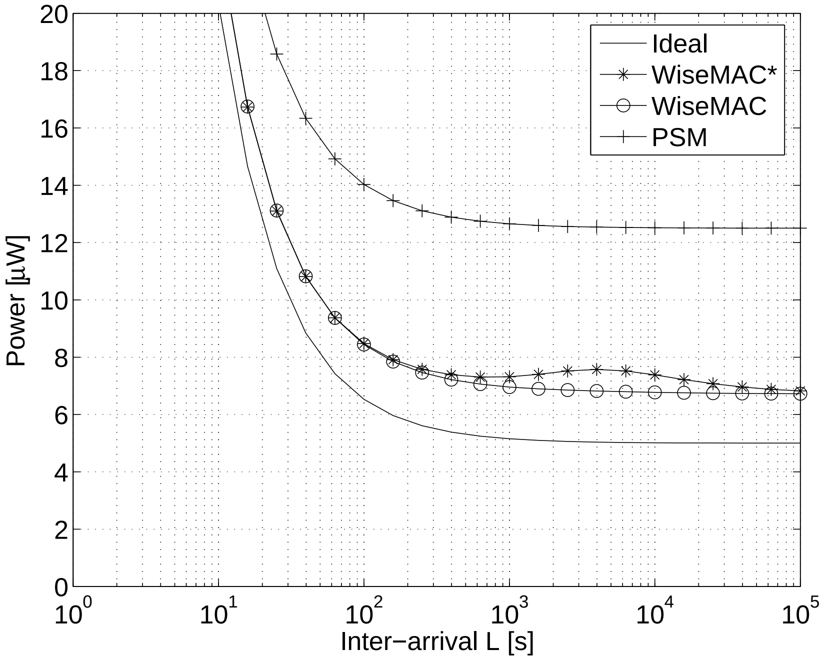

Figure 6.12: Power consumption as a function of the inter-arrival L when data frames are not repeated in the wake-up preamble ($T_W = 1$ s).

$$\overline{T}^*_{LP} = 2\theta L \left(1 - e^{-\frac{T_W}{4\theta L}}\right) \tag{6.20}$$

The average overhearing duration for a given preamble $T^*_O(T_P)$ was introduced in chapter 5, expression (5.6). With $T_W \gg T_D$, $T^*_O(T_P)$ can be approximated with $\frac{(T_P+T_D)^2}{2T_W}$ over the whole interval $0 < T_P \leq T_W$. The average overhearing duration can then be computed as $\overline{T}_O = \int_0^{T_W/4\theta} \frac{(T_P+T_D)^2}{2T_W} \frac{1}{L} e^{-\frac{l}{L}} dl + \int_{T_W/4\theta}^{\infty} \left(\frac{T_W}{2} + T_D\right) \frac{1}{L} e^{-\frac{l}{L}} dl$ which gives:

$$\overline{T}^*_O = \frac{1}{T_W}\left((4\theta L)^2 + 4\theta L T_D + \frac{T_D^2}{2}\right)\left(1 - e^{-\frac{T_W}{4\theta L}}\right) - 4\theta L e^{-\frac{T_W}{4\theta L}} \tag{6.21}$$

The power consumption of the WiseMAC* protocol is shown in Fig. 6.12 as a function of the inter-arrival L. It can be compared to the power consumption of WiseMAC, i.e. with repetitions of the data frame in the wake-up preamble. One can observe a local maximum in the power consumption of WiseMAC*. With periodic traffic, the maximum would be at $L = T_W/(4\theta) \approx$ 8000 s. With Poisson traffic, the maximum is at $L \approx 4000$ s. For inter-arrivals below this value, the length of the wake-up preamble is smaller than T_W and overhearing is mitigated in a probabilistic way. For inter-arrivals above this value, overhearing is not mitigated anylonger. Because data frames are not repeated in the wake-up preambles, overhearers have to listen to the wake-up preamble of every transmission during $T_W/2$ seconds on average. With increasing values of L, the power consumption of WiseMAC* converges again to the power consumption of WiseMAC, as the power consumption of data packet reception (including overhearing) becomes negligible.

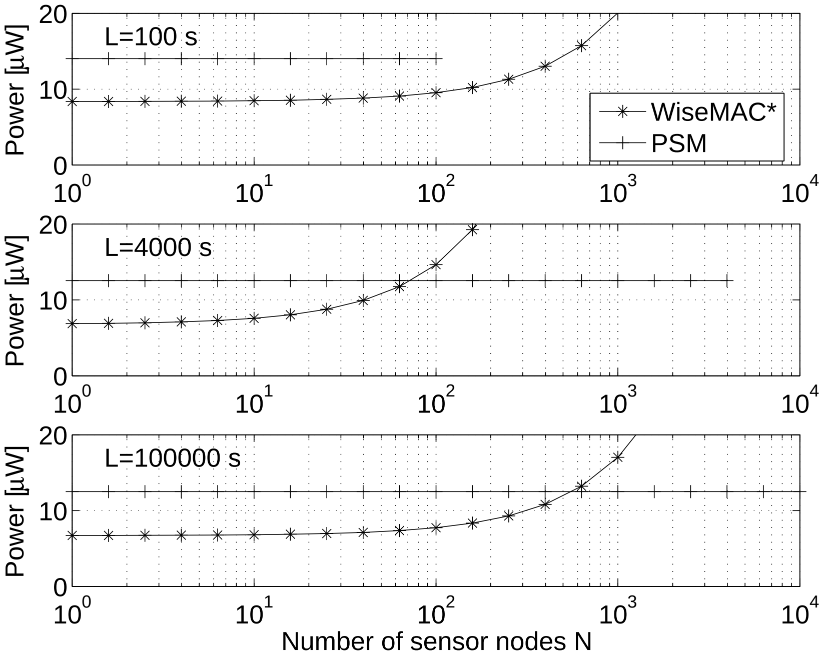

Figure 6.13: Power consumption as a function of the number of sensor nodes N, for different values of L ($T_W = 1$ s).

With $N = 10$ sensor nodes, one can see that WiseMAC* remains more energy efficient than PSM, for all values of L. However, as the overhearing component is proportional to $N - 1$ (the number of potential overhearers), WiseMAC* presents a scalability limitation, especially when the inter-arrival L is maximizing $\overline{T}_O^*$. This can be seen in Fig. 6.13, which shows the power consumption as a function of the number of nodes N (to be compared with Fig. 6.11). With a small and with a large inter-arrival ($L = 100$ s, $L = 100000$ s), the overhearing overhead remains small and WiseMAC* scales relatively well. However, if the average inter-arrival is equal to $L = 4000$ s, overhearing is maximized and WiseMAC* scales poorly. Note however that the scalability limitation appears for networks with more than 100 nodes, and that many applications will not require so many nodes.

6.6.4 Impact of the packet size

An increase or a decrease of the length of the data packet has no significant impact on the comparison of the protocols performance when the traffic is low. In high traffic conditions, having longer data packets will accelerate the convergence of WiseMAC and PSM towards the ideal protocol, as protocol overheads are less important with long data packets.

An increase of the length of control packets T_C has no impact on the power consumption of WiseMAC, but penalizes both PSM (due to the regular reception of a beacon of length T_C) and PTIP (due to the regular transmission of a poll packet of length T_C). A decrease of the length of control packets would bring the performance of PSM and PTIP closer to the one of WiseMAC.

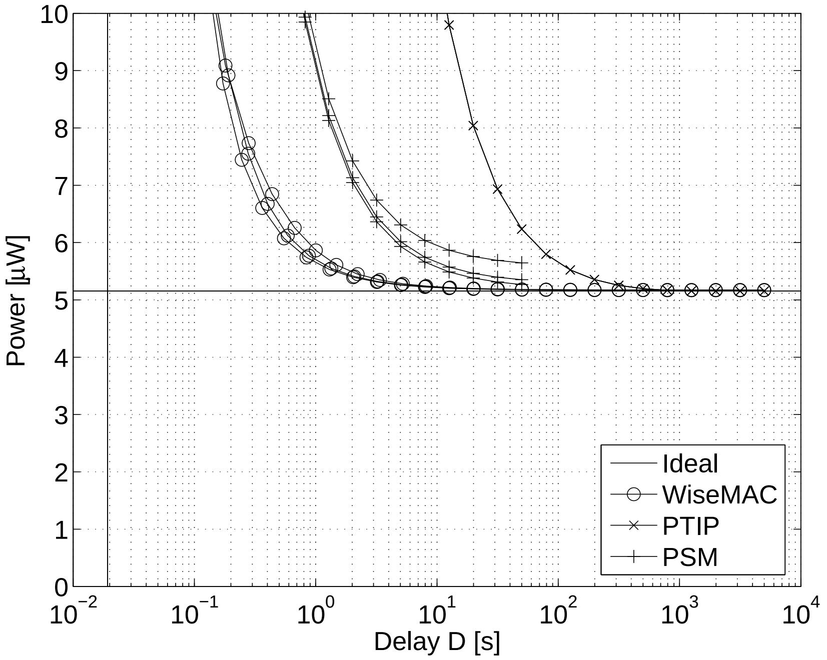

Figure 6.14: Power delay characteristics of WiseMAC, PTIP and PSM for $L = 1000$ s and $\theta = 10, 30, 100$ ppm. A larger power consumption corresponds to a larger θ.

6.6.5 Impact of the quartz frequency tolerance

We have considered a quartz frequency tolerance of $\theta = \pm 30$ ppm, a tolerance easily obtained with low cost 32 kHz watch crystals. Let us look at what happens when having a better or a worse accuracy.

First, one can notice that the PTIP protocol is not sensitive to the quartz accuracy. Sensor nodes may send poll packets at any time.

With the PSM protocol, sensor nodes must wake up in advance in order to cover the clock drift before to receive the beacon. This causes an average power consumption of $2\theta\widehat{P}_R$.

With the WiseMAC protocol, the length of the wake-up preamble is proportional to the frequency tolerance θ. Having a larger (resp. smaller) frequency tolerance will increase (resp. decrease) the overhearing overhead.

Fig. 6.14 shows the impact on the power consumption of WiseMAC and PSM, when using different values for the quartz tolerance $\theta = 10, 30, 100$. To map the curves on the values of θ, note that a larger θ causes a larger power consumption with both WiseMAC and PSM. It can be observed that the variations are small.

6.6.6 Impact of the TX/RX power consumption ratio

With the WiseNET transceiver, the ratio between the power consumption in transmit and in receive states is equal to $P_T/P_R = 35/2.1 = 16$. This ratio can be different depending on the transceiver and the chosen output power.

Fig. 6.15 shows the power consumption of WiseMAC, PTIP and PSM for a transmission power varying between $P_T = P_R$ and $P_T = 15P_R$. With large inter-arrivals, we have seen that the dominating components of the power consumption are the power consumption in doze

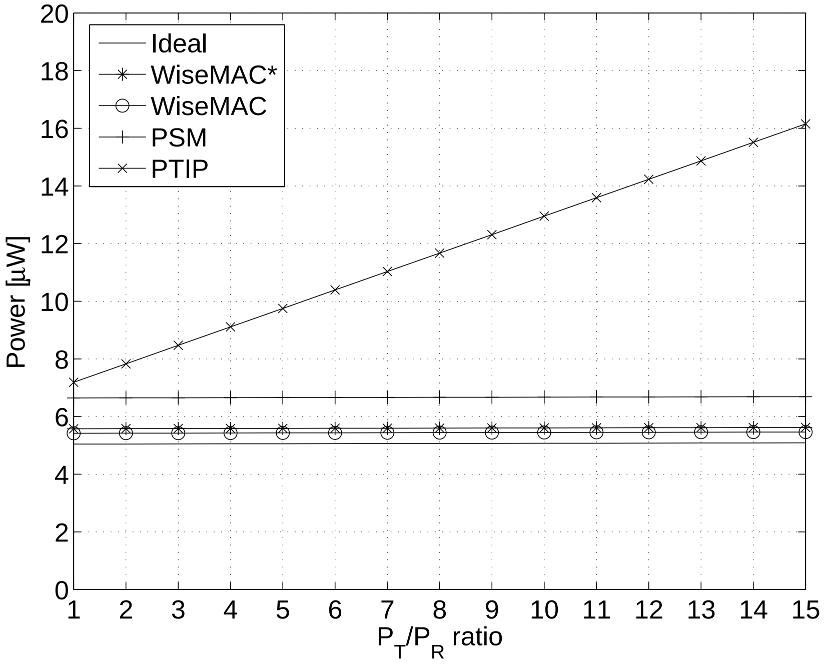

Figure 6.15: Power consumption of WiseMAC, PTIP and PSM for $L = 1000$ s as a function of the $\overline{P}_T/\overline{P}_R$ ratio. The chosen wake-up period is $T_W = 5$ s to remain in the validity domain of PTIP.

state and the power consumption of the wake-up scheme. With WiseMAC and PSM, the power consumption of the wake-up scheme does not depend on P_T. Their power consumption is hence almost constant when varying P_T. The comparison made between PSM and WiseMAC hence remains valid for any P_T/P_R ratio. On the other hand, the power consumption of PTIP depends on P_T. The smaller the ratio P_T/P_R, the closer the power consumption of PTIP and PSM. As mentioned earlier, the length of the TIM beacon may need to be larger than T_C. If the length of the TIM beacon is larger than the total length of a poll packet and of the control packet sent in reply, then PTIP may consume less power than PSM.

6.6.7 Impact of the bit rate

We have considered up to now the 25 kbps low bit rate WiseNET transceiver. In this section, let us analyze the impact of using high bit rate transceivers. We will consider the 11 Mbps Lucent Orinoco PC-Card and the 250 kbps Chipcon CC2420 802.15.4 transceiver. The parameters of these transceivers are listed in Table 6.1. The performance of the 3 protocols when using those transceivers is shown in Fig. 6.16.

The power consumption of the Lucent Orinoco card in the doze, receive and transmit modes is respectively $P_Z = 50$ mW, $P_R = 0.9$ W, $P_T = 1.4$ W. These values, as well as the bit rate R and the transmit power of 15 dBm, are taken from the product sheet [112]. The turn-around time T_T is assumed to be 5 μs, which is the maximum allowed value specified by the standard (aRxTxTurnaroundTime in [79]). The RSSI integration time is assumed to be $T_I = 15$ μs, the longest possible value according to the standard [79]. The setup time is estimated to be $T_S = 0.83$ ms. It has been estimated by measuring the duration of the current peak drawn by

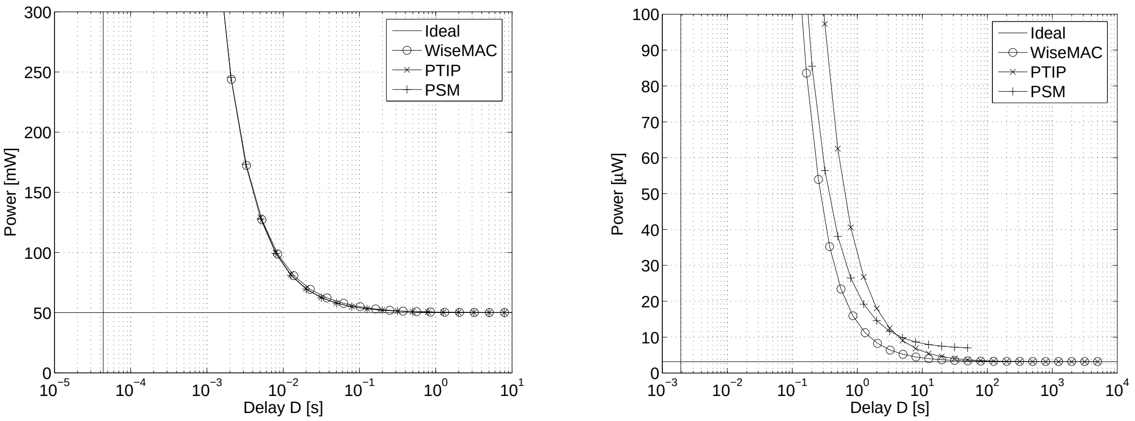

Figure 6.16: Power-delay characteristics of WiseMAC, PTIP and PSM when using a 11 Mbps Lucent Orinoco IEEE 802.11 transceiver (left) and a 250 kbps Chipcon CC2420 802.15.4 transceiver (right). The inter-arrival time is $L = 1000$ s.

Table 6.1: Transceiver Parameters

Lucent Orinoco	Chipcon CC2420
$P_Z = 50$ mW	$P_Z = 3\ \mu$W
$P_R = 0.9$ W	$P_R = 59.1$ mW
$P_T = 1.4$ W (@ 15 dBm)	$P_T = 52.2$ mW (@ 0 dBm)
$P_S = 0.93$ W	$P_S = 9.2$ mW
$T_S = 0.83$ ms	$T_S = 1.35$ ms
$T_T = 5\ \mu$s	$T_T = 192\ \mu$s
$R = 11$ Mbps	$R = 250$ kbps
$T_I = 15\ \mu$s	$T_I = 128\ \mu$s

the PCMCIA card during beacon reception (see Fig. 6.17). This peak has a width of 1.6 ms. Subtracting the duration of the PHY layer preamble (192 bits at 1 Mbps, i.e. 192 μs) and the duration of the beacon frame (72 bytes at 1 Mbps, i.e. 576 μs), gives a setup time of about 0.83 ms. The average power consumed during the startup phase $P_S = 0.93$ W was computed by taking the average over the first 0.83 ms in Fig. 6.17. At 11 Mbps, the length of a 60 bytes data packet becomes $T_D = 43\ \mu$s, and the length of a 12 bytes control packet becomes $T_C = 8.7\ \mu$s.

The parameters for the Chipcon CC2420 transceiver are taken from the data sheet [15]. Current values are multiplied with 3 V to obtain power values. It is assumed to power down the chip completely, including the voltage regulator. The current consumption is then 1 μA, which gives 3 μW with a 3 V supply. The setup time T_S is composed of 0.3 ms voltage regulator startup, 0.86 ms oscillator startup and 0.192 ms PLL startup, giving a total $T_S = 1.35$ ms. The chip consumes respectively 20 μA, 426 μA and 19.7 mA during the startup of the voltage regulator, the oscillator and the PLL. This gives a setup energy of 12.4 μJ and an average setup power of $P_S = E_S/T_S = 9.2$ mW. At 250 kbps, the length of a 60 bytes data packet becomes $T_D = 1.9$ ms, and the length of a 12 bytes control packet becomes $T_C = 0.4$ ms.

It can be observed in Fig. 6.16 that when using a IEEE 802.11 transceiver, the power consumption of all protocols is approximately identical. At 11 Mbps, the length of a control packet becomes very small (8.7 μs at 11 Mbps), as well as the energy to receive it. On the other hand, the setup energy remains relatively large. The advantage of WiseMAC over PSM and PTIP was

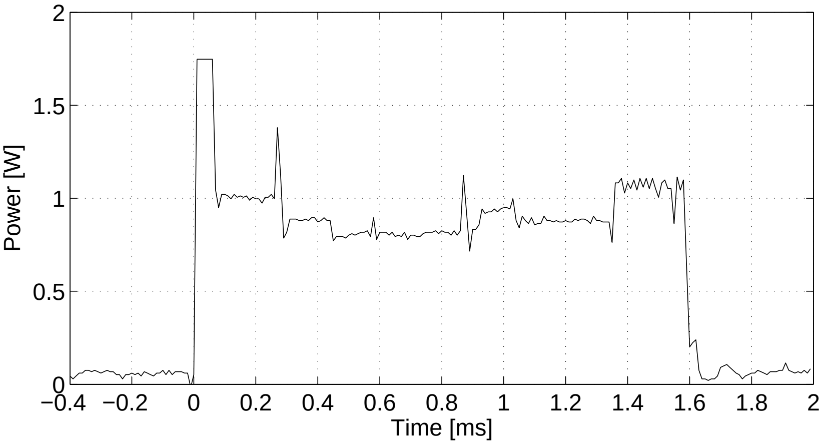

Figure 6.17: Power consumption of a Lucent Orinoco IEEE 802.11 PC Card during the reception of a beacon.

that sampling the medium costs less energy than receiving or transmitting a control message. If the energetic costs of sampling the medium and of receiving or transmitting a message is negligible compared to the setup energy, the power consumption of PSM and WiseMAC becomes identical. The periodic beacon used by PSM to inform sensor nodes of incoming traffic can serve other purposes, including time synchronization and access point discovery in roaming scenarios. The use of PSM for IEEE 802.11 networks was hence indeed a good choice.

With the IEEE 802.15.4 transceiver, the performance of the protocols are closer from one another than was observed with the WiseNET transceiver. As in the case of the IEEE 802.11 transceiver, the higher bit rate of the IEEE 802.15.4 transceiver (250 kbps) compared to the WiseNET transceiver (25 kbps) makes the advantage of WiseMAC smaller. Whether one should use WiseMAC or PSM with such a transceiver depends on the relative importance to a given application of the low power consumption and of the additional functions that PSM can bring with its beacon.

6.7 Conclusion

In this chapter, we have evaluated the performance of WiseMAC for the downlink of an infrastructure sensor network. A comparison was made with PTIP, a very simple protocol based on terminal initiated polling and PSM, the power save protocol used in the IEEE 802.11 and IEEE 802.15.4 standards. The trade-off between power consumption and delay was analyzed in low traffic conditions. Analytical expressions were derived to compute the power consumption and transmission delay of each protocol, as a function of the wake-up period.

It was shown that WiseMAC provides, with low bit rate radio transceivers, a significantly lower power consumption than PSM. Using the WiseMAC protocol with the WiseNET transceiver, a sensor node consumes 7 μW to receive 60 bytes data packets every 1000 seconds with an average latency of 0.5 seconds. When using the PSM protocol, reaching the same latency would cause a power consumption of 12 μW, almost two times more than with WiseMAC.

When the wake-up period can be chosen to be very large, it was seen that the power consumed by the wake-up scheme of all protocols becomes negligible. In such a case, the PTIP protocol becomes attractive as well, because of its implementation simplicity.

With high bit rate transceivers, it was seen that the power consumption of all protocols converge to the power consumption of the periodic wake-up. With an increasing bit rate, the additional activity specific to each protocol becomes negligible. Choosing PSM instead of WiseMAC becomes attractive because of the additional functions that a beacon can offer.

Chapter 7

Experimentation

7.1 Introduction

This chapter presents experimental results obtained with an implementation of the WiseMAC protocol. After a presentation of the used hardware platform, the operation of the WiseMAC protocol will be illustrated with oscilloscope traces. Average power consumption and delay measurements are finally presented.

7.2 Hardware platform

In order to advance in parallel with the finalization of the WiseNET SoC, the implementation of the WiseMAC communication protocol investigated in this dissertation was first made on a two-chips platform composed of the XE88LC06A [124] microcontroller and the XE1203F [127] FSK radio transceiver. Both chips are products from Xemics, a spin-off company of CSEM.

7.2.1 Microcontroller

The XE88LC06A [124] is an 8-bit microcontroller designed for wireless communication applications. It is based on the same CoolRISC 816 core [123] used in the WiseNET SoC, facilitating code porting towards the WiseNET SoC platform.

The architecture of the XE88LC06A microcontroller is illustrated in Fig. 7.1. The computer architecture of the CoolRISC 816 core is of the Harvard type [123], with a physically separated storage for instructions and data. The size of the instruction memory is of 8192 instructions. As instructions have a width of 22 bits, this corresponds to 22 kbytes. The size of the data memory is only 512 bytes. When programming in C with the gcc CoolRISC compiler, one must reserve a zone on the top of the data memory for the software stack, reducing further the space available for program variables.

When compared to the classical Von Neumann architecture [119], where data and instructions are stored in the same memory, the Harvard architecture presents the advantage of allowing the execution of one instruction per clock cycle. As power consumption is proportional to clock frequency, the Harvard architecture permits a lower power consumption per MIPS. The disadvantage of the Harvard architecture is to prevent the reading of constants directly from the program memory. Many algorithms (e.g. ciphers and CRCs implementations) are based on the use of lookup tables with 256 entries to map a byte onto the result of some complex operation.

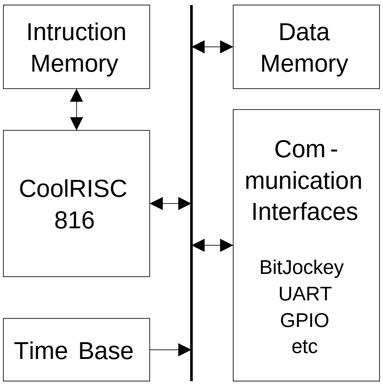

Figure 7.1: Simplified hardware architecture of a XE88LC06A microcontroller.

With a Harvard architecture and only 512 bytes of memory, such a lookup table would consume half of the data memory.

The communication interfaces of the XE88LC06A include a universal asynchronous receiver transmitter (UART), general purpose input/output pins (GPIO) and the BitJockey radio interface. The role of the BitJockey is to convert the serial interface towards the radio into a byte interface towards the CPU, relieving it from computing intensive operations such as bit and frame synchronization. In the transmit direction, bytes are transmitted as a serialized bit stream at the desired bit rate. An interrupt is generated when the transmit buffer is empty to inform the CPU when the next byte must be written. In the receive direction, the BitJockey recovers the bit clock, detects 8 bits patterns in the bit stream and generates interrupts when bytes have been received. In between the writing or reading of bytes, the CPU is able to perform other tasks and/or enter a power saving mode.

Two power-saving modes are available: stand-by and sleep. The stand-by mode is entered when issuing the "halt" command. It will also be referred to as the "halt" mode. In this mode, the CPU is stopped but the clocks remain active, allowing the peripherals to continue to work (e.g. time-keeping, UART and radio reception). The CPU is woken up by interrupts. In sleep mode, the clocks are stopped. Only an external reset can wake-up the CPU. As time-keeping is crucial to the WiseMAC implementation, this mode has not been used.

The CPU clocks can be generated by an internal RC oscillator and/or by an external 32 kHz quartz. The RC oscillator provides the high frequency clock necessary for high speed processing and the 32 kHz quartz provides the high accuracy time-keeping base.

7.2.2 Radio transceiver

The XE1203F [127] is a FSK transceiver designed for the 433, 868 and 915 MHz bands. It provides a sensitivity of $\mathcal{P}_S = -109$ dBm at a bit rate of 25 kbps and a bit error rate of 10^{-3}. The bit rate can be selected between 1.2 and 152.3 kbps. The transmitted power can be configured to be 0, 5, 10 or 15 dBm. In order to match the WiseNET SoC parameters, a bit rate of $R = 25$ kbps and a transmission power of $\mathcal{P}_T = 10$ dBm will be used.

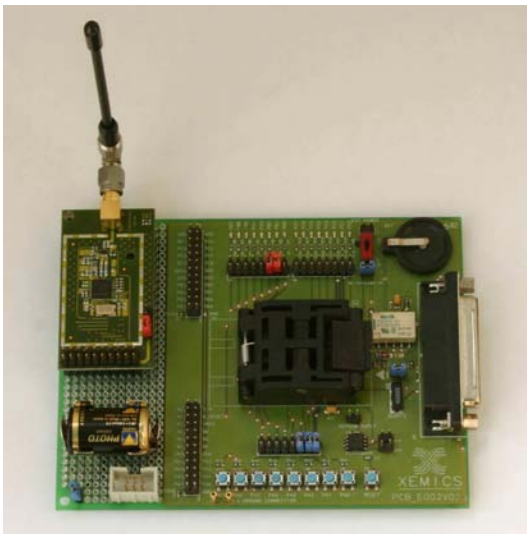

Figure 7.2: EV108 Development board with XM1203 radio module.

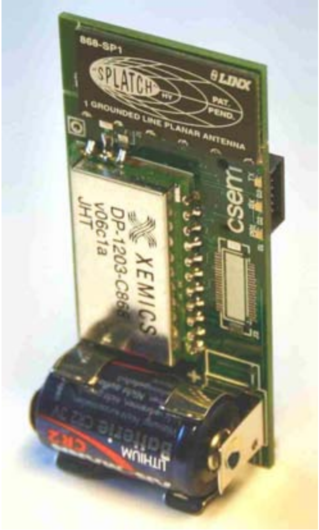

Figure 7.3: WiseNode: A miniaturized low power wireless sensor node based on the XE1203 radio and the XE88LC06A microcontroller.

7.2.3 Development and demonstration boards

The measurements of the current consumed by the XE88LC06A microcontroller and the XE1203F transceiver have been made using a XM1203 module operating at 868 MHz driven by the XE88LC06A micro-controller on a EV108 [128] evaluation board (see Fig. 7.2). The XM1203 module includes a XE1203F chip and the required additional radio frequency components (39 MHz quartz, antenna switch, etc.). The EV108 board includes a socket for the XE88LC06A microcontroller, a 32 kHz quartz, buttons and LEDs. An integrated sensor node device using the same chips has been realized (see Fig. 7.3).

7.3 Software architecture

The communication software located on the sensor node implements routing and the MAC layer. Low level APIs have been defined to interface the radio transceiver and the time-keeping subsystem, allowing software compatibility between platforms at MAC layer and above.

If the computing resources requirements of an application are low, it will be possible to include the application software on the same microcontroller as the communication software (see Fig. 7.4, left). If the processing or memory requirements are too large, a second microcontroller must be used for the application software (see Fig. 7.4, right). In such a case, it is foreseen that the two will communicate through a serial interface called the Host Controller Interface (HCI, after the Bluetooth terminology [102]). Following the Bluetooth HCI specification, the host controller sends commands and receives events. Events may be answers to the commands or generated asynchronously due to, for example, the reception of a packet. Commands and events are formatted using a TLV (type, length, value) scheme.

The size of the different software components is shown in Fig. 7.5. In its current version, the implementation of the WiseMAC protocol requires 1606 words of 22 bits, totalizing 4.5 kbytes.

The remaining 5008 words (14 kbytes) are used by the radio driver (XE1203Driver), the UART driver (UartDriver), the RC frequency calibration routine (DFLLDriver), the time-keeping driver (CntDriver), the HCI interface (Hci), the routing layer (Route), the initialization and scheduling routines (crt0 and Main) and the basic gcc support routines such as floating point operations and modulo operations (libgcc_sc).

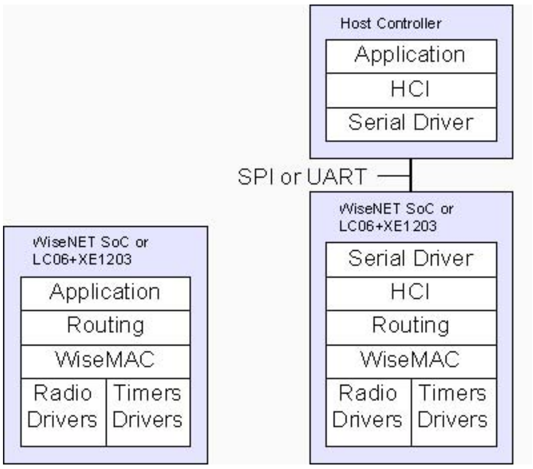

Figure 7.4: Single (left) and dual (right) controller solution.

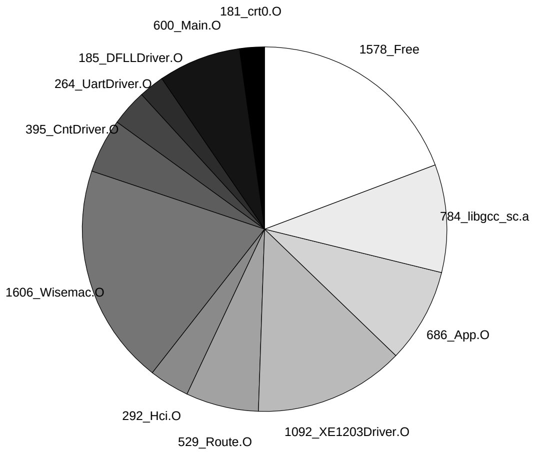

Figure 7.5: Code size in words of the different software components.

7.4 Measurements

7.4.1 Time-keeping base accuracy

Table 7.1 shows the measured oscillation frequency of the 32 kHz quartz on 5 EV108 boards and on five WiseNode integrated sensor nodes. Measurements have been made at 20 degrees centigrade. These measurements show that the assumption, made in this thesis, to considering a quartz tolerance θ within $[-30; +30]$ ppm is valid at room temperature with these quartzes. Temperature variations and aging will increase the inaccuracy of the quartzes. It was shown in section 5.4.5 that a larger quartz tolerance results in a higher power consumption. To improve the synchronization accuracy in such cases, a node may have to learn the drift rate of its neighbors and record it along with the sampling offset.

7.4.2 Static current consumption

The measured static current consumption of the microcontroller and the radio transceiver are shown in table 7.2.

The most common activity of a sensor node will be to sleep. To minimize the power consumption in doze state, it is important to use the 32 kHz quartz as the clock source, and to switch off the radio oscillator and the CPU RC oscillator. The doze power consumption becomes $P_Z = 2.4 \cdot 3 = 7.2$ μW. The power consumption in receive and transmit state (10 dBm output power) are respectively $P_R = 14 \cdot 3 = 42$ mW and $P_T = 43 \cdot 3 = 129$ mW.

Table 7.1: Effective 32kHz quartz oscillation frequency

Board	Frequency	Deviation
EV108 No 1	32767.3 Hz	-21 ppm
EV108 No 2	32767.6 Hz	-12 ppm
EV108 No 3	32767.5 Hz	-15 ppm
EV108 No 4	32767.8 Hz	-6 ppm
EV108 No 5	32767.6 Hz	-12 ppm
WiseNode No 1	32768.9 Hz	+27 ppm
WiseNode No 2	32768.5 Hz	+15 ppm
WiseNode No 3	32768.5 Hz	+15 ppm
WiseNode No 4	32768.8 Hz	+24 ppm
WiseNode No 5	32769.0 Hz	+30 ppm

Table 7.2: Measured static current consumption of XE88LC06A and XE1203 (3 V).

CPU halt 32 kHz, RC off, radio in sleep mode	2.4 μA
CPU halt 32 kHz and RC 2.4 MHz running	84 μA
CPU running 32 kHz (Quartz)	12 μA
CPU running 2.4 MHz (RC)	760 μA
Radio oscillator running	1.0 mA
Radio RX	14 mA
Radio TX 0 dBm	24 mA
Radio TX 5 dBm	30 mA
Radio TX 10 dBm	43 mA
Radio TX 15 dBm	55 mA

7.4.3 Dynamic current consumption

The dynamic current consumption during the transitions between the doze, receive and transmit states has been measured by recording with an oscilloscope the voltage drop across a 1 Ohm resistor. Fig 7.6 shows the current consumed when following the radio model state machine (Fig. 3.3 in chapter 3) clockwise (doze-receive-transmit-doze) and anti-clockwise (doze-transmit-receive-doze). The thick line shows the current consumed by the radio module, while the light line shows the total consumed current. The difference between the two is due to the power consumption of the micro-controller and to the charge and discharge of decoupling capacitors.

The vertical dotted lines indicate when the commands are starting to be transmitted by the micro-controller to the radio transceiver. There is a delay between the time the command is issued and the time it is executed on the radio transceiver. This delay is caused by the time needed to send the command to the radio transceiver using a 3 wires serial interface implemented in software on a general purpose parallel port. It amounts to 140 μs, which corresponds to the time needed to execute the function WriteRegister (Xemics XE120x API [125]) at 2.4 MHz CPU frequency.

Following the transceiver datasheet [126], waiting times of 500 μs and 150 μs have been programmed after respectively receiver and transmitter start-up. The oscillator needs 500 μs to start. A waiting time of 360 μs has been programmed after issuing the oscillator start command. Sending the following command will give the additionally required 140 μs.

The duration of the setup phase into receive state amounts to $T_{SR} = 1.14$ ms. This duration includes the initial 140 μs to send the oscillator start command, 360 μs waiting time, 140 μs to

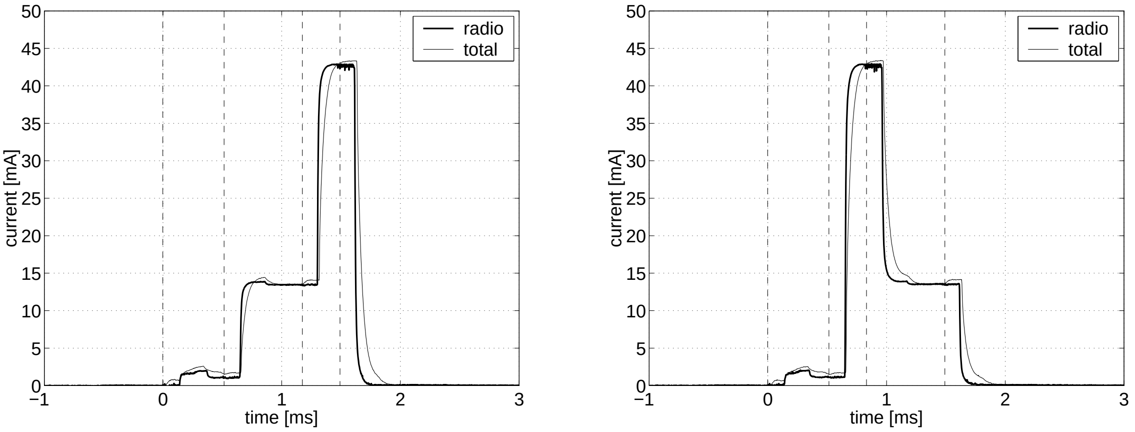

Figure 7.6: Current consumption of a XE1203F radio driven by a XE88LC06A micro-controller when traveling clock-wise (left, doze-receive-transmit-doze) and anti-clockwise (right, doze-transmit-receive-doze) in the transceiver state machine.

Table 7.3: XE88LC06A and XE1203F two-chips solution.

$P_Z = 7.2\ \mu$W	P_{SR}= 19 mW	$T_{SR} = 1.14$ ms	$\mathcal{P}_T = 10$ dBm
$P_R = 42$ mW	$P_{ST} = 40$ mW	$T_{ST} = 0.79$ ms	$\mathcal{P}_S = -109$ dBm (BER=10^{-3})
$P_T = 129$ mW	$P_{TR} = 42$ mW	$T_{TR} = 0.64$ ms	$R = 25$ kbps
$U = 3$ V	$P_{RT} = 129$ mW	$T_{RT} = 0.29$ ms	$T_I = 640\ \mu$s

send to command requesting the switch into receive state followed by 500 μs waiting time. The energy consumed for setup into receive state integrates to 22 μJ at 3 volts. This corresponds to an average power $P_{SR} = 19$ mW during T_{SR}.

The duration of the setup phase into transmit state amounts to $T_{ST} = 0.79$ ms. This duration includes the initial 140 μs to send the oscillator start command, 360 μs waiting time, 140 μs to send to command requesting the switch into transmit state followed by 150 μs waiting time. The energy consumed for setup into transmit state integrates to 32 μJ at 3 volts. This corresponds to an average power $P_{ST} = 40$ mW during T_{ST}.

To obtain a fast switching between states, it is possible with the XE1203F transceiver to program in advance the next state and to switch into this state by toggling a signal on an input pin. This optimization could not yet be implemented in the software used in these tests and a delay of 140 μs is added to the turn-around and turn-off procedures. The duration of the turn-around phases is then $T_{RT} = 290\ \mu$s in the receive to transmit direction, and $T_{TR} = 640\ \mu$s in the transmit to receive direction. As can be seen in Fig 7.6, the power consumption during the turn-around phases quickly settles to the power consumption of the destination state. One can make the approximation $P_{TR} = P_R$ and $P_{RT} = P_T$.

The RSSI integration time is fixed to 500 μs. After the measurement delay, reading the value through the 3 wires serial interface causes an additional delay of 140 μs. The total time needed to obtain a measurement is hence $T_I = 640\ \mu$s .

The parameters of the XE88LC06A and XE1203F two-chips solution are summarized in table 7.3.

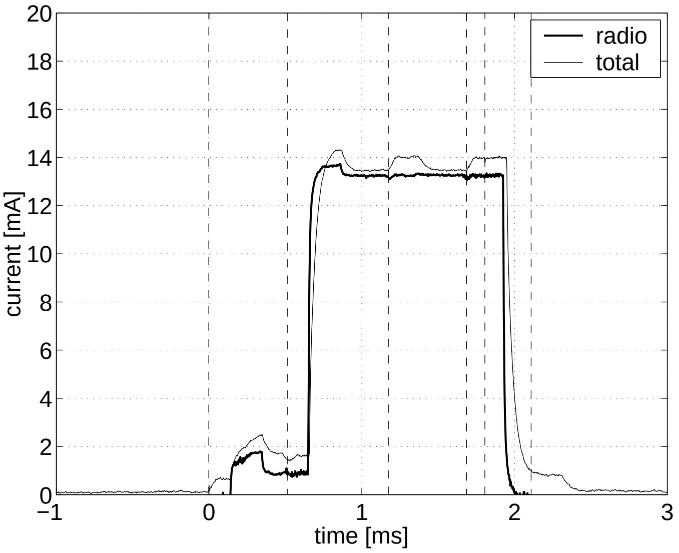

Figure 7.7: Current consumed by the XE1203 radio transceiver (thick line) and total consumed current (XE1203 and XE88LC06A) when sampling the medium.

7.4.4 Energy consumption of sampling

Figure 7.7 shows the current consumption during the sampling procedure. The two first vertical lines represent respectively when the commands to turn-on the oscillator and the radio are starting to be transmitted. The RSSI is measured between the third and the fourth vertical lines. The increase in the total current consumption at the beginning of the RSSI measurement phase is due to the programming of the timer counting the 500 μs RSSI integration time. The next interval represents the time needed to read the RSSI value. The last interval shows the time needed to stop the radio.

With the radio model introduced in section 3.3 and using the parameters values given in table 7.3, the energy needed to sample the medium is given by $P_{SR}T_{SR} + P_RT_I = 19 \cdot 1.14 + 42 \cdot 0.64 = 48.5$ μJ. The numerical integral of the measured current consumption gives an energy of 52.3 μJ for the radio and 55 μJ in total. The difference of 3.8 μJ between the energy consumed by the radio and the prediction by the model stems from the energy consumed between the time when the turn-off command is sent and the time when it is received by the radio. The model assumes that an instantaneous turn-off is possible. With an optimized implementation using the fast switching feature of the transceiver, these few μJ could be saved. The implementation would then match the model accurately.

It is interesting at this point to compare the 48.5 μJ needed by an off-the-shelf transceiver operating at 3 V to sample the medium with the $P_{SR}T_{SR} + P_RT_I = 0.4 \cdot 1.7 + 2.1 \cdot 0.1 = 0.89$ μJ needed by the WiseNET SoC operating at 1 V. The sampling activity consumes 54 times less with the WiseNET SoC, making possible the use of a high sampling rate while keeping a low average power consumption.

7.4.5 Minimization of the wake-up preamble length

Fig. 7.8 shows the current consumption and the transceiver state of a source and a destination when using the WiseMAC protocol. The current is measured using a 1 Ohm shunt. The state of

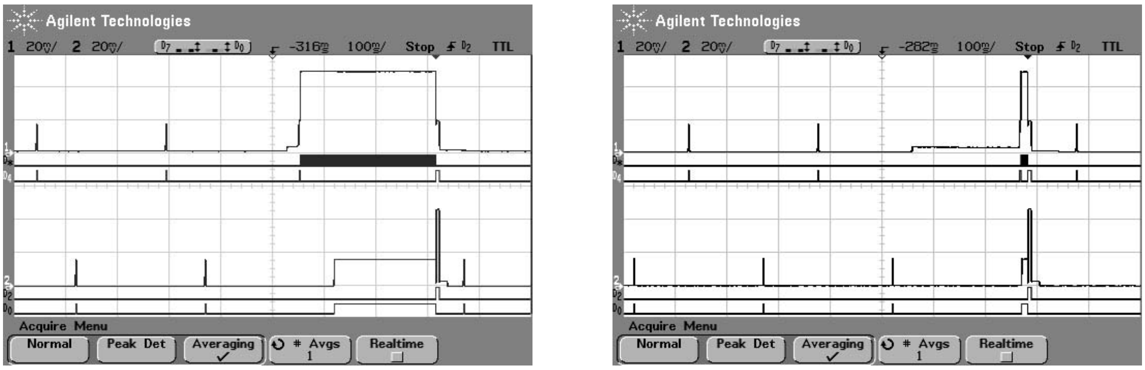

Figure 7.8: Transmission with wake-up preamble of the length of the sampling period (left) and transmission with a wake-up preamble of minimized size (right) ($T_W = 250$ ms).

the transceiver is observed through the monitoring of two pins indicating when the transceiver is respectively in receive or in transmit state. When both signals are low, the transceiver is sleeping. In Fig. 7.8, the three upper curves belong to the source, while the three lower belong to the destination. In each group, the first one represents the consumed current, the second one the transmit state and the third one the receive state.

The short peaks on the current consumption and on the receive state indication lines represent the sampling activity. It can be observed that source and destination are not sampling the medium at the same times. Their sampling schedules are independent.

The transmission from the source is generated at a random time through the manual activation of a push-button. At the packet generation time, the microcontroller exits the low power sleeping state and activates the RC. To mitigate collisions between packets that might be sent by multiple nodes as a result of an external event (detected event, received broadcast message), the WiseMAC medium access control requires a random waiting time before to attempt a transmission with a long preamble. As a reminder, transmissions with a minimized preamble are not preceded by a backoff, but include a medium reservation preamble of randomized size in front of the wake-up preamble. The backoff waiting time can be seen in Fig. 7.8 (left) through the small current consumption increase due to the RC oscillator that is preceding the transmission. In this example, the backoff time amounts to about 20 ms. Just before to start the transmission, the source enters the receive state to perform carrier sensing. After the transmission, which lasts for about 260 ms, the source turns its transceiver around into the receive state, waiting for the acknowledgement packet.

The destination detects the wake-up preamble and stays in receive state until the data packet is received. The repetition of the data packets in the wake-up preamble is not used in this example. After the correct reception of the data packet (as attested by the frame check sequence), the acknowledgement packet is transmitted.

Once the acknowledgement received, the source knows the offset between its sampling schedule and the one of the destination. For the next transmission, it can use a wake-up preamble of minimized size. The right part of Fig. 7.8 shows a transmission with a wake-up preamble of minimized size. Again, the packet is generated at a random time through a push-button. At the packet generation time, the RC oscillator is started, and the next suitable transmission time is computed. The waiting time between the arrival of the packet and the start of the transmission

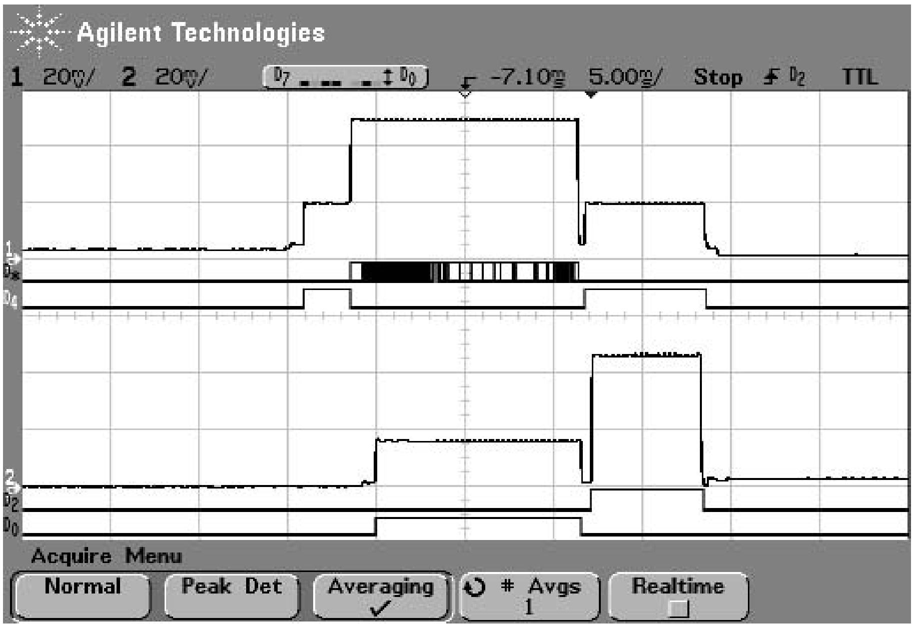

Figure 7.9: Detailed view of a transmission with a wake-up preamble of minimized size.

can be observed in Fig. 7.8 through, as in the previous case, the increased current consumption due to the RC oscillator. Prior to transmit, the source enters the receive state to perform carrier sensing. The data packet is then transmitted and answered with an acknowledgement. A detailed view of the transmission with a short wake-up preamble is shown in Fig. 7.9. In this example, the wake-up preamble has a size of about 4 ms. The wake-up preamble has been made visible by superimposing to the transmit state indicator, the signal on the serial data path from the micro-controller towards the transceiver. As the wake-up preamble is composed of alternating bits, it is visible as the black area at the front of the packet.

7.4.6 Multi-hop transmission

In many sensor network applications, a packet will have to be relayed in a multi-hop fashion. The forwarding of a packet across three nodes is illustrated in Fig. 7.10. The tree groups of signals correspond to the three nodes. On each node, a buffer stores the packets received from the radio or from the upper layer until they are transmitted over the air or given to the upper layer. The first line in each group indicates when the packet buffer on a node contains a data packet. The second line indicates when the transceiver is in transmit state, and the third one when the transceiver is in receive state. Through the observation of the first line in each group, one can see the packet moving from the packet buffer of a node to the next one. Once arrived at destination, the packet is given to the upper layer and the packet buffer is emptied.

Observing the receive and transmit state indication signals, one can recognize the carrier sensing operation before the transmissions as well as the transmissions of data and acknowledgement packets.

It can be observed that only the destination of a transmission is woken up by the transmission. This demonstrates the overhearing mitigation effect brought by the independence between the sampling schedules of the different nodes combined with the minimization of the wake-up

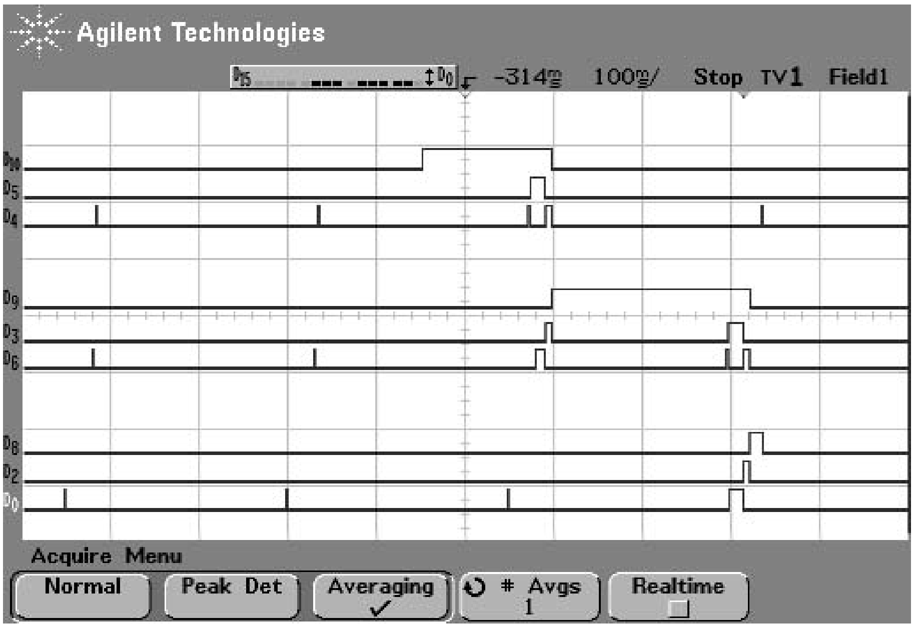

Figure 7.10: Multi-hop transmission of a packet ($T_W = 250$ ms).

preamble.

In this example, the total delay between the arrival of the packet in the first node and its delivery to the upper layer on the third node amounts to 380 ms. The hop delay is hence 190 ms, slightly below the wake-up period $T_W = 250$ ms.

7.4.7 Average power consumption and transmission delay

The average power consumption has been measured indirectly through recording the time spent by the radio transceiver in the receive and transmit states. A direct measurement of the average current consumption was not possible due to the large integration time required, of up to hundreds of seconds. The micro-controller on each node is used to sum up the time spent by the radio in the receive and transmit state. The granularity of the time measurement is equal to a oscillation period of the 32 kHz quartz, which amounts to about 30 μs. The power consumption is computed by multiplying the average time spent in doze, receive and transmit states by respectively P_Z, P_R and P_T as given in table 7.3.

The measurement setup is illustrated in Fig. 7.11. Traffic is generated by node 1 and forwarded by the 4 other nodes before coming back to the source node. Packets are sent with an average interval of L seconds. The instant when the packet is generated is randomized using an initial backoff equal to the sampling period. Routing tables have been pre-configured such that packets follow the depicted route. The advantage of having a circular traffic is to permit to simultaneously measure the average power consumption and the transmission delay. When a packet is generated, the source node introduces a time stamp in its payload. The transmission delay is computed by comparing this time stamp with the time at arrival. The power consumption is measured on the source node. Its power consumption is equal to the one of the forwarders as it sends and receives one packet every L seconds.

Fig. 7.12 shows the average power consumption obtained with the WiseMAC protocol as a

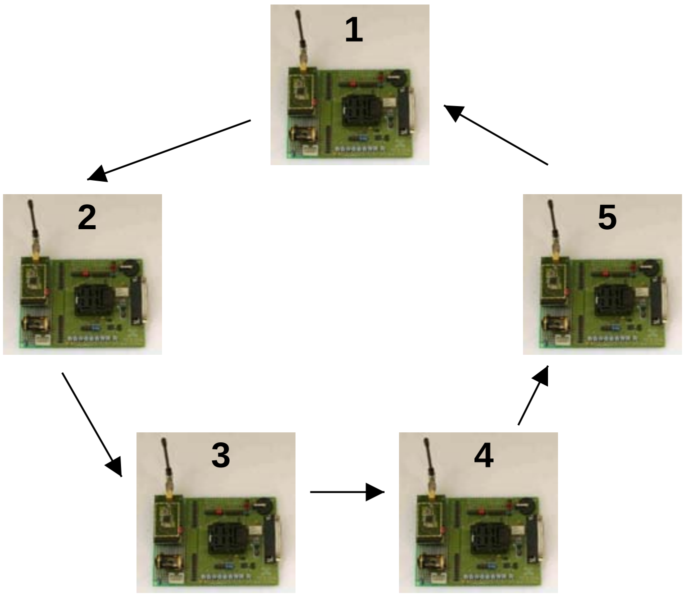

Figure 7.11: Topology used for average power consumption and average delay measurements.

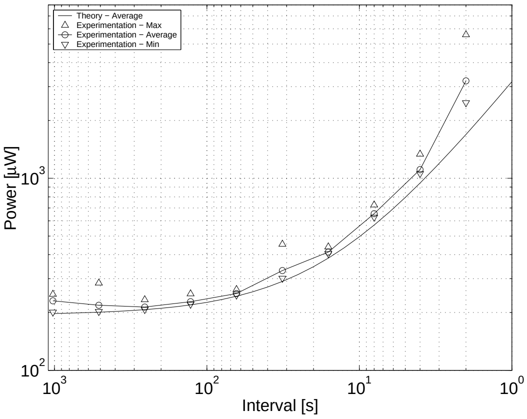

Figure 7.12: Average power consumption, computed from the measured time spent with the radio in receive and transmit states (T_W = 250 ms).

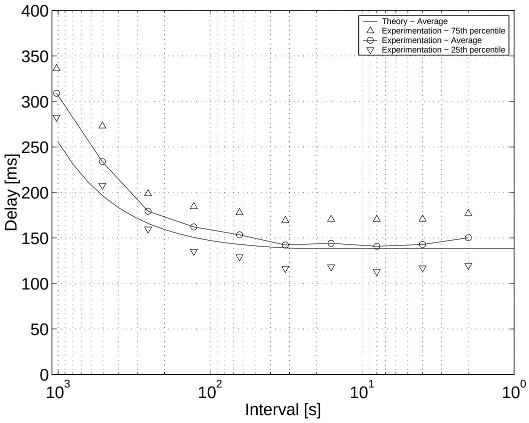

Figure 7.13: Average hop transmission delay measured in a circular network composed of 5 nodes ($T_W = 250$ ms).

function of the inter-arrival time and with $T_W = 250$ ms. The line with circles markers shows the average power consumption measured through experiments, while the plain line shows the theoretical power consumption predicted by expression (5.19) using the radio parameters of the XE88LC06A-XE1203F two-chips solution given in Table 7.2. Six experiments were run for each traffic intensity. For each experiment and traffic intensity, 20 packets are transmitted, the average power is computed. The average of these six average values is represented by the circle markers. The triangle markers show the minimum and the maximum average power consumption over these six experiments. It can be observed that the measured average power consumption is close to the theoretical average power consumption.

Fig. 7.13 shows the average hop transmission delay, obtained through a measurement of the round trip delay of packets traveling in circle across the 5 nodes. The solid line with circle markers shows the average over 120 packets (20 packets for each of the six experiments). The triangle markers show the 25th and 75th percentiles of the delay distribution. The solid line without markers shows the theoretical average delay as predicted by expression (5.23).

If all the nodes were sampling the medium in a synchronized way the round trip delay of a packet would consist first of the waiting time between the packet generation and the transmission time. With $T_W = 250$ ms, this waiting time would be $T_W/2 = 125$ ms in average. It would then need 4 additional hops to come back to the source node, giving an additional delay of $4T_W = 1$ s. The total delay would hence be 1.125 ms, giving a hop delay of 225 ms. As the sampling times of the nodes are independent, the delay at each hop is smaller than 250 ms. When the traffic is sufficiently large for the wake-up to be small, an average hop delay of about 125 ms can be measured. When the traffic becomes small (e.g. one packet every 2 minutes, or every 4 minutes), the hop delay augments because of the need to send relatively long wake-up preambles in front of data packets. This behavior was already observed with the simulation model of WiseMAC. In the simulation results, one could also observe an increase of the delay when the inter-arrival time was below $L = 10$ s. This delay increase in high traffic conditions is not visible in the results of this experiment. The reason of this better performance is that there are only 5 nodes forwarding a packet in the used setup, while in the simulated network, 10 groups of 10 nodes were forwarding traffic in parallel, resulting in a higher overall traffic.

7.5 Conclusion

The implementation of the WiseMAC protocol was carried out on a two-chip platform based on the XE88LC06A microcontroller and the XE1203F radio.

First, it was demonstrated that it is possible to implement the WiseMAC protocol on a very memory limited platform (22 kbytes instruction memory and 512 bytes data memory). The software module implementing the WiseMAC layer uses less than 5 kbytes of instruction memory.

The experimentations have validated the basic mechanisms introduced in the WiseMAC protocol, which are the usage of preamble sampling at the destination side, the adaptive learning of the offset between a destination and a source and the usage of a wake-up preamble of minimized length at the source side. It was seen that the measurements of the average power consumption and of the average hop delay as a function of the traffic behave as predicted by theory.

Chapter 8

Conclusion

This thesis deals with energy efficient medium access control protocols designed to meet the specific requirements of wireless sensor networks. The main requirements of wireless sensor networks were identified to be a low cost and a long lifetime. Additional requirements are the self-organization, the scalability and the capability to operate in a multi-hop fashion.

The protocols investigated in this dissertation are based on the preamble sampling technique, which is a form of periodic reception. This technique, used in paging systems, had received up to now little attention in the research community. This thesis has provided a detailed mathematical analysis of the performance of a protocol combining non-persistent CSMA with the preamble sampling technique. The renewal theory available for the analysis of classical non-persistent CSMA has been extended to cover the case with preamble sampling and to provide power consumption information.

The strength of the preamble sampling technique is to put a very low power consumption overhead at the destination side, giving to the source side the responsibility and the cost of waking up a destination. With the basic preamble sampling technique, a wake-up preamble of the size of the sampling period must be used, which leads to a low energy efficiency in high traffic conditions.

The WiseMAC protocol, introduced in this dissertation, builds on CSMA with preamble sampling technique. By piggy-backing the remaining time until the next sampling instant in each acknowledgement packet, WiseMAC permits to each node to learn and refresh the offset between its own sampling schedule and the ones of its destinations. In addition, WiseMAC makes use of a number of mechanisms to mitigate collisions and overhearing. These mechanisms include the extension of the carrier sensing range to mitigate the hidden node effect, the use of mandatory inter-frame spaces to avoid the interruption of data-acknowledgement transactions, the addition of a medium reservation preamble of randomized size in front of wake-up preambles of minimized size to mitigate systematic collisions, the transmission of bursts of packets using the "more" bit, and finally, the repetition of data frames in long wake-up preambles to mitigate overhearing. The performance of the WiseMAC protocol was analysed mathematically and through simulation, and compared to other state-of-the-art protocols.

It was shown the WiseMAC can simultaneously achieve a low average power consumption in low traffic conditions and approach the energy efficiency of TDMA in high traffic condition. The fact that the energy efficiency of TDMA is approached with WiseMAC is remarkable given the simplicity of the WiseMAC protocol as compared to the complexity of managing a spatial TDMA schedule.

When using the WiseNET SoC, a node will consume only 28 μW to forward a packet of 60 bytes every 100 seconds. A lifetime of 5 years could then be reached using a single AA alkaline battery. The WiseMAC protocol was shown, at that traffic, to be 75 times more energy efficient than the classical non-power saving CSMA/CA and 8 times better than S-MAC, operated at its default duty cycle of 10%.

An implementation of the WiseMAC protocol on a low-power embedded platform has demonstrated the practical validity of the proposed mechanisms. It was shown that this protocol can be implemented with a memory footprint of a few kbytes only, as required to meet the low cost requirement. Measurements of the average power consumption and of the average hop delay have confirmed results provided by the mathematical analysis and by simulations.

The WiseMAC protocol was shown to be an attractive MAC protocol, not only for wireless multi-hop sensor network, but also for the downlink of infrastructured based wireless sensor networks, allowing the use of a single scheme for communication in a hybrid topology.

The WiseMAC protocol was designed to operate on a single frequency. A direction for further development could be to target the support of multiple frequency channels, in order to increase the system capacity and be able to avoid interferers through frequency agility.

At the physical layer, further work will be needed to evaluate the trade-offs between the used bit rate, the transmission power, the modulation and the channel coding schemes in the context of low duty cycle wireless sensor networks.

At the network layer, further work include the evaluation of routing protocols suitable for operating above the WiseMAC layer. The WiseMAC protocol is based on a flat topology, as opposed to a clustered topology with for example the IEEE 802.15.4 protocol [82]. This characteristic may help in designing simple routing algorithms. With WiseMAC, broadcast transmissions must be of the duration of the sampling period to wake up and reach every node in range. Broadcast is used by many multi-hop routing protocols such as DSR [51] or directed diffusion [49]. As using long wake-up preambles is energetically costly, the usage of broadcasts will negatively impact power consumption. The support of mobility will require broadcast messages for network discovery. A high mobility will be associated with a relatively high power consumption. However, many sensor network applications are expected to be quasi static. In such cases, the energetic cost of self-organization can be expected to be low. When mobility is desired, solutions to implement an energy efficient routing on top of the WiseMAC protocol remain to be designed.

Congestion is a source of collisions and energy waste. Flow control is traditionally implemented between end-points at the transport layer. In a multi-hop sensor network, it may be of interest, as proposed by Woo in [122], to insert a ”stop” bit in the MAC layer acknowledgements. Such a mechanism is compatible with the WiseMAC protocol. Its performances remain to be evaluated in conjunction to WiseMAC. This scheme may permit to inform more rapidly the sources of a congestion than classical transport layer schemes.

A further potential source of energy savings can be found in the field of mobile code. Mobile code may save energy, for example, in applications where distributed processing would require more energy to transfer the information to be processed than to transfer the algorithm to be applied to the data.

Appendix A

Interference Between Bluetooth Piconets

A.1 Introduction

Bluetooth is a digital wireless data transmission standard in the 2.4 GHz ISM band aimed at providing a short range wireless link between laptops, cellular phones and other devices [102]. The air interface modulation is Gaussian FSK with a raw bit rate of 1 Mb/s. The communication topology between Bluetooth nodes is point-to-multipoint, where a master communicates in time division duplex with several slaves forming a so-called piconet. Even slots are used for packets from the master to one of the slaves and odd slots are used for the return direction. In order to tolerate interference which can readily arise in the 2.4 GHz band, a slow frequency hopping scheme is used, where all nodes of a piconet hop together among 79 frequencies at each packet slot. As Bluetooth is meant to be used for applications such as connecting a headset to a mobile phone, the problem of co-channel interference from other Bluetooth piconets can become of high importance. It is likely to have several persons in proximity, each having an open Bluetooth connection between a mobile phone and a headset or a mobile computer. Simulation results addressing the problem of interference between Bluetooth networks can also be found in [137] and [136].

A.2 Model

We assume n unsynchronized collocated piconets that are sufficiently close from one another such that a co-channel interference between 2 or more packets will destroy all packets. For simplicity, we assume that only 1 bit of overlap is enough to destroy all packets involved. The interference system model is shown in Fig. 1. Forward error correction and capture effect are neglected. Because of the strong adjacent channel rejection requirement imposed by the standard, adjacent channel interference is not considered. The traffic in each piconet is assumed periodic and the packet rate is G packets per slot. G can take values between 0 and 1 and can be seen as the probability to have a packet in a slot. With $G = 1$, we have 100% traffic in each piconet.

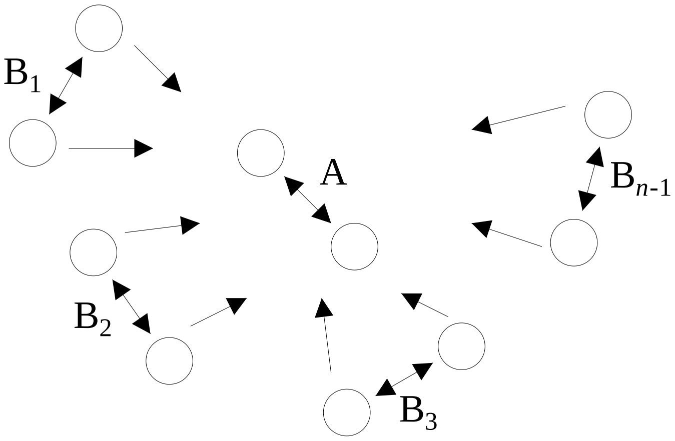

Figure A.1: Interference between piconets.

A.3 Packet Error Rate

We will first consider piconets that are synchronized among them. The probability that two nodes in piconet A can successfully exchange a packet when one other synchronized piconet B is collocated, is equal to $P_S^S = 1 - GP_1$, where $P_1 = 1/79$ is the probability that piconet B chooses the same frequency as the one chosen by piconet A. In the variable P_S^S , the superscript S stands for synchronized (or for simple, as will be seen later) and the subscript S for success. With n collocated piconets, the piconet A has $n - 1$ adversary piconets. The probability to successfully transmit is then

$$P_S^S(n) = (1 - GP_1)^{n-1} \tag{A.1}$$

According to the Bluetooth standard, different piconets are not synchronized. Another important point is that the duration of a packet is smaller than the duration of a slot. A single slot packet is 366 bits long ($T_P = 366$ μs at 1 Mbps) and the duration of the slot is $T_S = 625$ μs. The idle time between the end of the transmission of the packet and the start of the next reception is used to let the electronics stabilize at the next frequency. Let us consider again only two piconets. Depending on the relative time phase, one or two slots from the adversary piconet B can interfere with the packet of interest in piconet A. We assume that the time shift between A and B is a random variable uniformly distributed between 0 and T_S. The time shift is random, but it is constant for any given pair of piconets assuming that the slow clock drifts are neglected. Using the diagram shown in Fig. A.2, one can see that there is a probability $d = 2r - 1$ that the time shift is such that a packet in piconet B is a potential danger to 2 packets in piconet A. The probability that the time shift is such that a packet in piconet B is a potential danger to 1 packet in piconet A is the complement

$$s = 1 - d = 2(1 - r) \tag{A.2}$$

These values can be understood as follows: over all possible shifts of the adversary piconet, there are two zones of single exposure with a total length of $2(T_S - T_P)$ and one zone of double

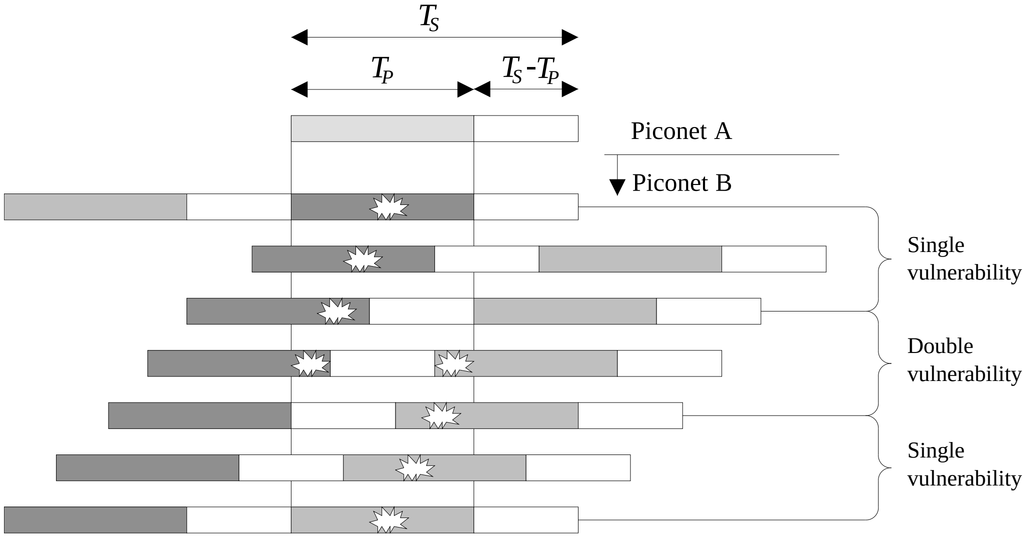

Figure A.2: Single or double exposition to interference.

exposure of length $T_P - (T_S - T_P) = 2T_P - T_S$. Normalized with T_S and using $r = T_P/T_S$, we obtain $2(1-r)$ and $2r-1$.

If the adversary piconet B is shifted such that one of its packets is threatening two slots in piconet A, the probability of transmitting successfully one packet is given by $P_S^D = (1 - GP_1)^2$, as both the preceding and the following slots must have chosen another frequency, in the case that they had a packet to send[1]. In the variable P_S^D, D stands for double exposure. Now, if there are n collocated piconets, and if all $n-1$ adversary piconets are shifted such that each of the packet sent by them is doubly threatening a packet sent in piconet A, the probability to successfully transmit becomes

$$P_S^D(n) = (1 - GP_1)^{2(n-1)} \tag{A.3}$$

In a real situation, from the $n-1$ adversary piconets, there will be a number n_S of piconets simply threatening and a number $n_D = n - 1 - n_S$ of piconets doubly threatening. These numbers are random but constant over time for any given set of active piconets. Let denote with N_S the random variable taking individual values n_S. The distribution of N_S is binomial:

$$\mathsf{P}(N_S = n_s) = \binom{n-1}{n_S} s^{n_S} (1-s)^{n-1-n_S} \tag{A.4}$$

where s is defined in equation (A.2). For any given n and n_S, the probability of successfully transmitting a packet can be expressed as

---

[1] Another way to see this is to consider that the packet will suffer from interference and be in error if both dangerous slots choose the same frequency and have a packet to send (probability $(GP_1)^2$), or if the first one chooses the same frequency and have a packet to send and the second doesn't (probability $GP_1(1 - GP_1)$). This last event can be reversed. Therefore, the probability to have a packet error is $P_E^D = 2GP_1(1 - GP_1) + (GP_1)^2 = 2GP_1 - (GP_1)^2$ and $P_S^D = 1 - P_E^D = (1 - GP_1)^2$.

$$P_S(n, n_S) = (1 - GP_1)^{n_S}(1 - GP_1)^{2(n-1-n_S)} = (1 - GP_1)^{2(n-1)-n_s} \tag{A.5}$$

This expression is a lower bound on the packet success probability as a function of n and n_S. The packet success probability can be higher in a real system thanks to the capture effect and forward error correction codes. With $n_s = n - 1$, we obtain the synchronized case, i.e. formula (A.1). With $n_S = 0$, we obtain the double threat case, i.e. formula (A.3). The mean value of this lower bound over all possible shift configurations can be obtained by taking the expectation of (A.5) over all values of N_S:

$$\begin{aligned}
&P_S(n) = E\left[P_S(n, N_S)\right] \\
&= \sum_{n_S=0}^{n-1} (1 - GP_1)^{n_S} (1 - GP_1)^{2(n-1-n_S)} \left(\begin{array}{c} n-1 \\ n_S \end{array} \right) s^{n_S} (1 - s)^{n-1-n_S} \\
&= \sum_{n_S=0}^{n-1} \left(\begin{array}{c} n-1 \\ n_S \end{array} \right) (s(1 - GP_1))^{n_S} \left((1 - s)(1 - GP_1)^2\right)^{n-1-n_S} \\
&= \left(s(1 - GP_1) + (1 - s)(1 - GP_1)^2\right)^{n-1}
\end{aligned}$$

where we made use of the expansion formula for the Newton's binomial

$$(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

Replacing the variable s by T_P/T_S, we get

$$P_S(n) = \left(2\left(1 - \frac{T_P}{T_S}\right)(1 - GP_1) + \left(\frac{2T_P}{T_S} - 1\right)(1 - GP_1)^2\right)^{n-1} \tag{A.6}$$

A simpler way to find the same result, valid when $G = 1$, is given in [24]. Defining $a_M = s(1 - GP_1) + (1 - s)(1 - GP_1)^2$, we obtain $P_S(n) = a_M^{n-1}$, where M stands for mean. We see that in the three considered scenarios, the expression for the packet success probability has the form

$$P_S(n) = a^{n-1} \tag{A.7}$$

where a depends on the scenario. In the one considered at last, for $G = 1$, we have $a = a_M = s(1 - P_1) + (1 - s)(1 - P_1)^2 = 0.9852$. If we would have considered unsynchronized piconets but would not have taken into account the fact that the packet length is smaller than the slot length, we would be in the situation where $n_D = n - 1$, with $a = a_D = (1 - P_1)^2 = (1 - 1/79)^2 = 0.9748$. This corresponds to a_M with $s = 0$. With synchronized piconets, the ratio between packet and slot length is not relevant. We have always only one dangerous slot in an adversary piconet. The value of the variable a is then $a = a_S = 1 - P_1 = 1 - 1/79 = 0.9873$. This corresponds to a_M with $s = 1$. Assuming n unsynchronized piconets all transmitting with a packet rate $G = 1$, the packet error rate suffered by one piconet because of the interference from $n - 1$ adversary

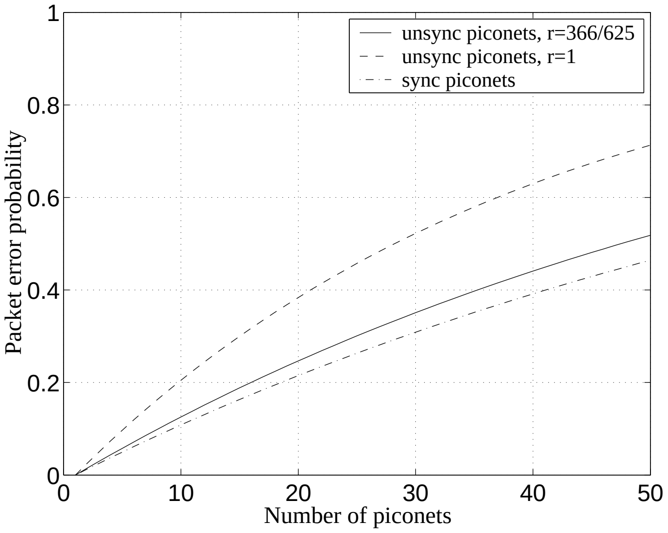

Figure A.3: Packet error rate suffered by one piconet because of the interference from $n-1$ adversary piconets.

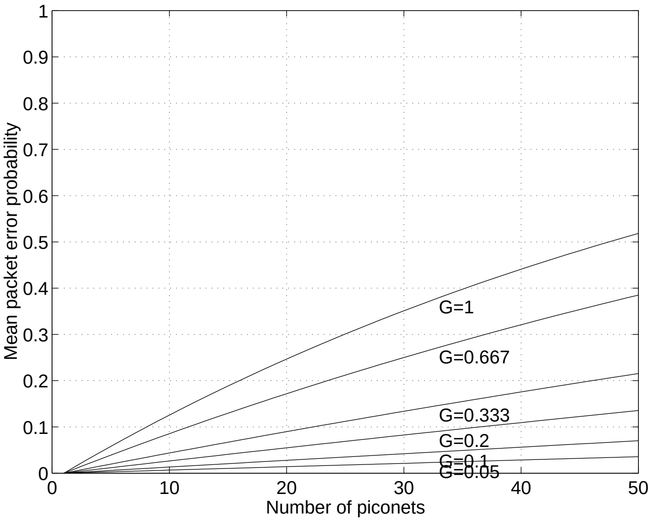

Figure A.4: Packet error rate suffered by one piconet because of the interference from $n-1$ unsynchronized adversary piconets ($r = 366/625$) for different values of the traffic G.

piconets is $P_E(n) = 1 - P_S(n) = 1 - a^{n-1}$. The plot of this function for the three different values of a can be seen in Fig. A.3. The curve in the middle shows the mean of the packet error probabilities over the possible distributions of the piconets in the groups of the single overlap and double overlap.

A lower traffic in the adversary piconets will reduce the packet error probability. The mean packet error probability $P_S(n)$ for different values of the packet rate G is plotted in Fig. A.4.

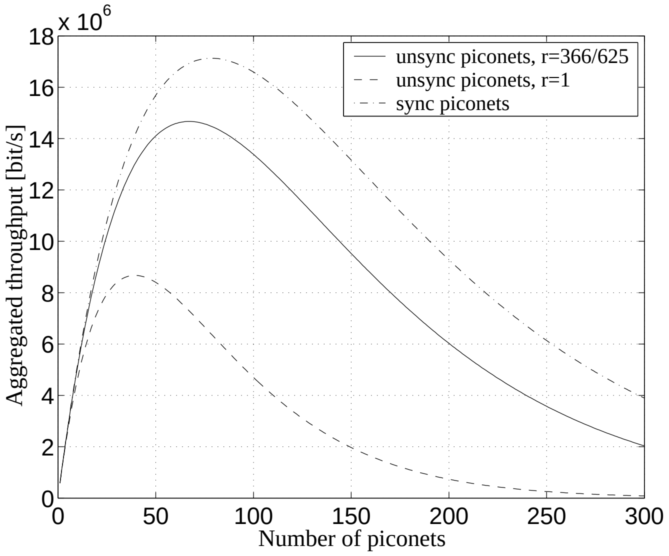

Figure A.5: Aggregated throughput.

Table A.1: n maximizing the aggregated throughput.

Case	a	$n = -1/\ln a$
async $r = 1$	$a = (78/79)^2$	39.2
async $r = 466/625$	$a = 0.9852$	67.1
sync	$a = 78/79$	78.5

A.4 Aggregated Throughput

Assuming n unsynchronized piconets all transmitting with a packet rate G of 1, the aggregated throughput of the successfully transmitted packets in all piconets can be expressed as

$$S_A(n) = n \cdot P_s(n) = n \cdot a^{n-1} \tag{A.8}$$

The plot of this function for the three different values of a can be seen in Fig. A.5.

The curve in the middle shows the mean of the aggregated throughput over the possible distributions of the piconets in the groups of the single overlap and double overlap. The unit conversion from packets/slot into Mbps is done by a multiplication with the factor 366 bits/625 μs. The maximum aggregated throughput is reached for $(n \cdot a^{n-1})' = 1 \cdot a^{n-1} + n \cdot a^{n-1} \ln(a) = 0$, which gives $n = -1/\ln a$. For the different curves, the maximum is reached for the value of n as given in Table A.1.

A.5 Simulation Model in OPNET

In order to validate the theoretical results, the interference scenario has been modeled using the Opnet network simulator [83]. The network topology that has been used is shown in Fig. A.6. The node bt_tx_node is sending a packet every 625 μs, and the node bt_rx_node is receiving these packets. These two nodes model a bi-directional link between a master and a slave in a

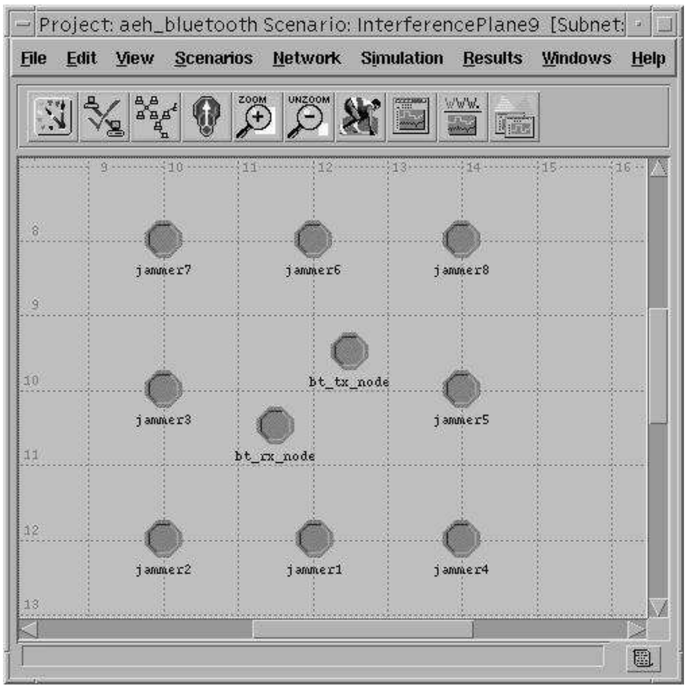

Figure A.6: Network topology with 9 interfering piconets.

piconet. The jammers nodes located around the central pair of nodes represent the interfering piconets. These nodes are simply transmitting data, one packet every 625 μs, at a frequency selected uniformly out of the 79 ones. No reception is computed. The number of jammers will be varied for the different simulations, to compare the results with the theoretical expressions given earlier as a function of n. The basic idea under this topology is that people carrying a mobile phone and a laptop or a headset will keep a certain distance between themselves.

It has been chosen to have jammers spaced by about 2 meters from each other, and from the central piconet. If the number of nodes is increased, it is the size of the area populated with jammer that is increased and not the density of the jammers. For example, Fig. A.7 shows the topology of a network with 49 jammers.

The source node is implemented using the state machine shown in Fig. A.8. At each BEGSLOT_INTRPT interrupt, the process chooses randomly one frequency among the 79 Bluetooth frequencies, it then transmits a packet of 366 bits at the selected frequency, schedule a BEGSLOT_INTRPT for itself and for the sink process 625 μs later. The packet is sent at 1 Mbps using a frequency band of 1 MHz.

The sink is implemented using the state machine shown in Fig. A.9. At each BEGSLOT_INTRPT interrupt, the process reads in a global variable what is the frequency that the source has selected for the current slot, sets the receiver module at that frequency, and waits for the reception of a packet. At packet reception, the Opnet simulator computes the interference that this packet has to face because of the jammers and allocate errors. If the number of errors present in the packet is above a threshold, the packet is dropped. This threshold has been set to zero, such that even one error in a packet causes its dropping. Hence, the optional forward error correction capabilities are not taken into account.

The jammer process model shown in Fig. A.10 is first composed of a wait state, where each

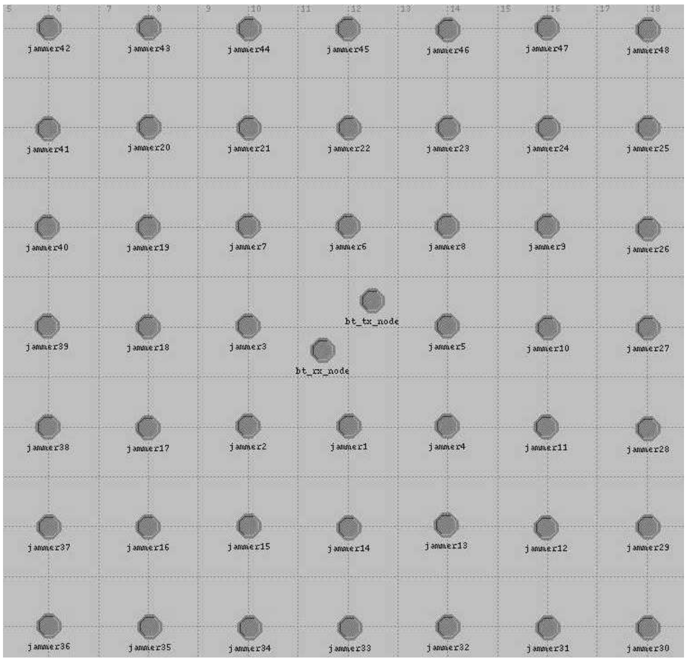

Figure A.7: Network topology with 49 interfering piconets.

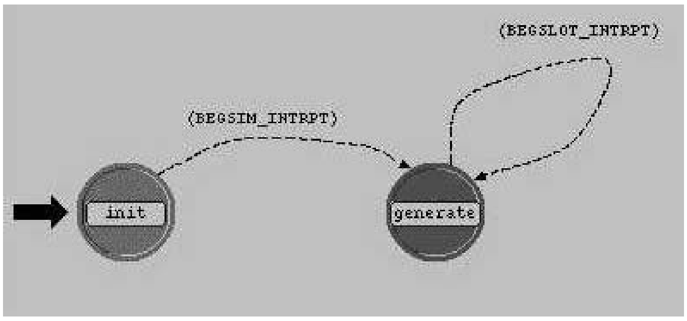

Figure A.8: Source process model.

jammer waits for a random time, uniformly distributed between 0 and 625 μs. The execution reaches then the generate state, where a frequency is selected at random, a packet is sent, and an interrupt scheduled to wake-up the process 625 μs later.

The propagation model that has been used in the simulation, is the free space attenuation $L_P = \lambda^2/(4\pi d)^2$ (the default model in Opnet). When allocating the errors, in the received packet, Opnet computes the signal to noise ratio and insert errors according to the bit error probability versus E_b/N_0 curve. For a GFSK receiver, this curve would be dependent on the receiver technology. Because such results were not yet available, the performance curve of non-coherent FSK modulation $P_b = \frac{1}{2}e^{-E_b/(2N_0)}$ has been used instead [105]. This deviation of the model from the Bluetooth system specification remains acceptable for this set of simulations, as we mostly want to verify the formulas for the extreme case, when the jammers are so close to the receiver that the signal to noise ratio is extremely bad anyway. For the scenarios where the nodes are further away, the results should not be considered as absolute values valid for the Bluetooth system.

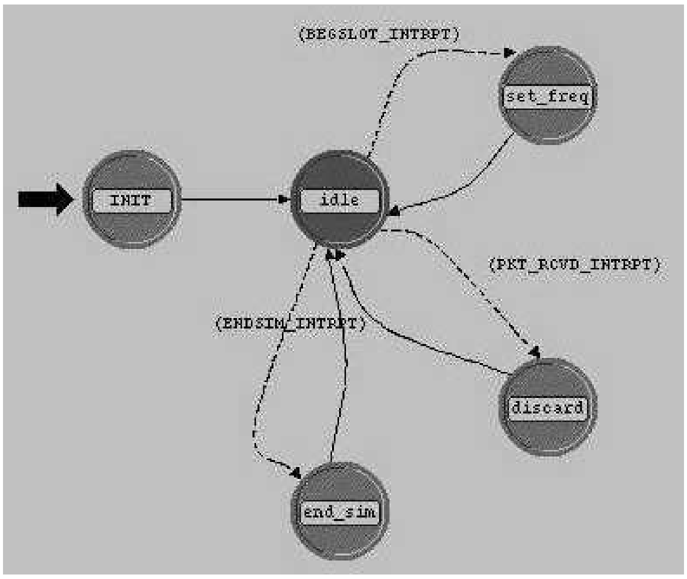

Figure A.9: Sink process model.

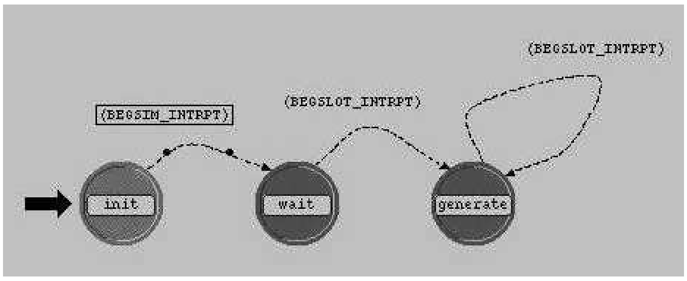

Figure A.10: Jammer process model.

A.6 Simulations Results

The simulations have been run for 6 scenarios, with $n = 5, 9, 17, 25, 37, 39$ jammers. The duration of the simulation in simulated time was 40 seconds for $n = 5, 9, 17, 25$, 60 seconds for $n = 37$ and 80 seconds for $n = 49$. For each scenario, 30 simulation runs have been performed using different seeds for the random number generator, resulting in 30 different selections of the time offset in each jammer.

The first set of simulations has been performed with a transmission power of 20 mW for the bt_tx_node and 2 W for the jammers. The higher power at the jammers is used to model collocated piconets in order to match the assumptions made for the derivation of the theoretical results. Fig. A.11 shows the packet error rate in function of the number of piconets (which is the number of jammers + 1).

The "x" signs results from simulations where the time offset has been forced to zero (i.e. synchronized piconets). We see that there is a good match with formula (A.1). The "o" signs results from simulations where the time offset has been forced to 313 μs (i.e. double threat from each jammer). We see that they match formula (A.3). The "+" signs results from simulations where the time offset at the jammer is randomly chosen at the beginning of the simulation.

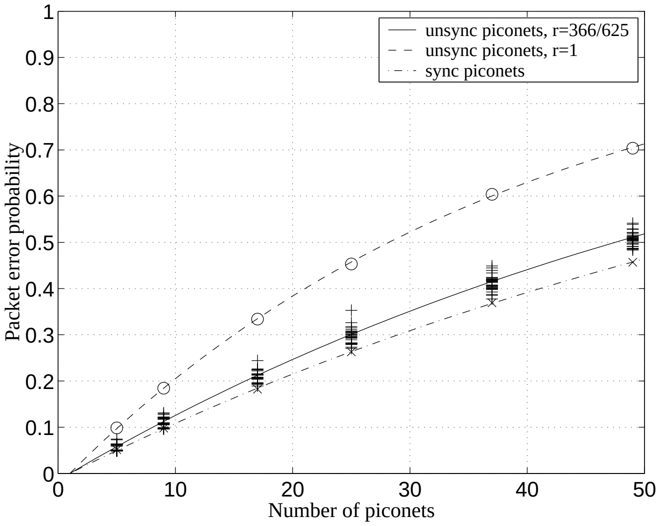

Figure A.11: Simulation results for jammers transmitting with a power of 2 W.

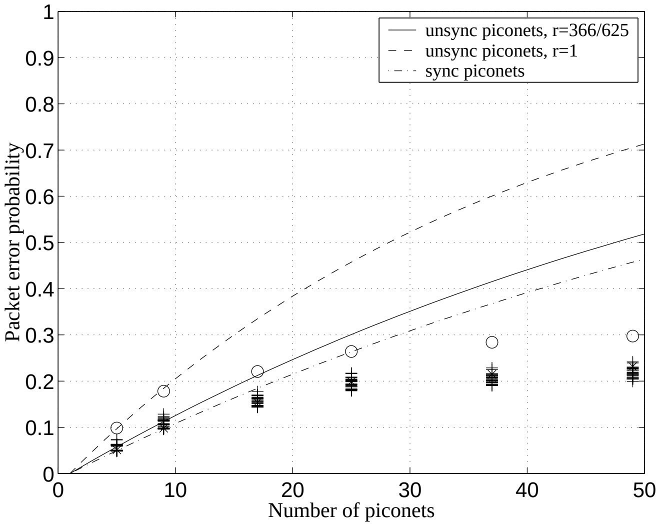

Figure A.12: Simulation results for jammers transmitting with a power of 20 mW.

Because the random choice of the time offset is likely to generate a number of simply and doubly threatening jammers that is close to the mean, these clouds of points are around the mean probability of error given in formula (A.6).

The second set of simulations has been performed with a transmission power of 20 mW for both the bt_tx_node and the jammers. The results, shown in Fig. A.12, give an idea of the impact of the capture effect. When the number of piconets becomes larger, more and more jammers are far away. Because these additional jammers are far away, their contribution to the degradation of the signal to interference ratio becomes negligible. Therefore, the packet error probability remains almost constant as n increases.

A.7 Conclusion

The analysis presented here gives an upper bound on the mean packet error rate of a Bluetooth link under co-channel interference from $n - 1$ other piconets, as well as a lower bound on the aggregated throughput of n collocated piconets. In a more general way, this analysis provides some insights to the problem of co-channel interference between Bluetooth piconets. It permits also to quantify the potential gain that one would obtain if different piconets would be synchronized by some means external to the Bluetooth standard. These expressions, as they do not take into account the effect of capture and the possible correction of errors using forward error correction, can be used to measure these effects in terms of packet error probability reduction. Simulation results illustrate this probabilistic analysis and show the potential effect of capture on the packet error probability.

Appendix B

Simulation Model

B.1 Simulation plateform

Because of the complex nature of multi-hop wireless sensor networks, there is a strong need to perform simulations of newly designed communication protocols, before to attempt implementation and real world testing.

At the time of the project start (in 2001), there existed only few discrete event network simulators that could be used for wireless MAC protocol simulation[1]: OPNET [83], NS-2 [113] and Glomosim [134].

OPNET is a very popular commercial tool, where protocols are implemented in pseudo-C language inside finite state machines. NS-2 is widely used among routing and transport protocols researchers, mainly in the field of TCP/IP wired networks. A wireless component has been added to enable research on adaptive routing for wireless ad hoc networks. Protocols are implemented in C++ and simulations scenarios are defined with Object Tcl scripts. Glomosim is a simulator dedicated to the simulation of wireless networks. It aims at providing a short execution time of simulation involving tens of wireless nodes. Glomosim has been developed on top of the discrete event simulation language "Parsec". This language is composed of a very limited number of commands (mainly send, send after some delay and hold). The core of the software is the scheduler, which is passing messages among nodes and continuously reordering the list of future events. Building upon simple constructs, the Glomosim software has evolved towards a complex simulator offering an OSI layered structure, with a number of different ad hoc routing and wireless medium access control protocols. Because of its processing efficiency, its clean software structure and its open source character, we have decided to use and extend this simulator. Another advantage of Glomosim is that it is written in C, a language that is likely to be used for implementations on embedded sensors, thereby allowing a potential reuse of parts of the MAC layer model source code for the implementation.

B.2 Interference and radio layer simulation model

The radio layer of Glomosim did not include the modeling of low power functions. This model had to be completed with the addition of a doze state, where packets cannot be received, as

---

[1] Newly developed simulation platforms include SENSE [108] and JiST/SWANS [5]. Note that Glomosim has evolved into the commercial QualNET simulator [97]. The open source Glomosim simulator has not been developed any further since year 2000.

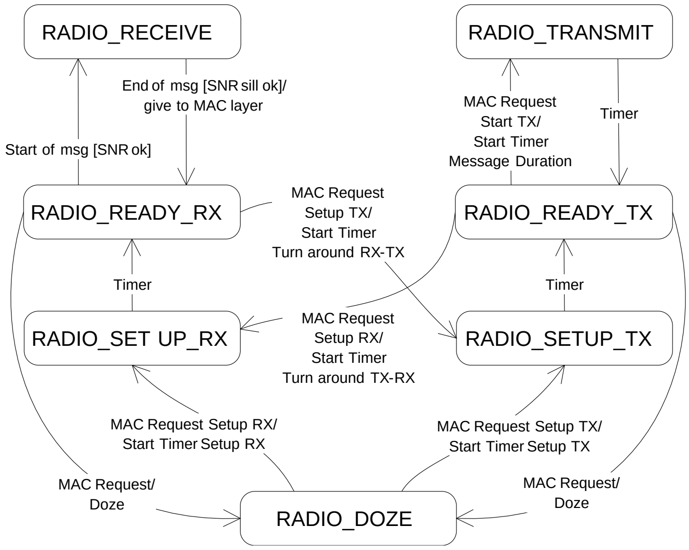

Figure B.1: Transceiver simulation model

well as the modeling of the setup and turn-around delays between the states doze, receive and transmit. A very detailed modeling of the temporal behavior of the radio transceiver is a fundamental step for the precise evaluation of the consumed power and for the correct choice of time parameters in the MAC protocol. In addition, a precise temporal behavior of the low layers is a prerequisite to any evaluation of synchronization protocols, which will for sure be needed in many sensor networks to provide time stamping.

The finite state machine of the model is shown in Fig. B.1. The syntax of the finite state machine is following the UML standard: *Event*[*Condition*]/*Action*. A transition is made if the *Event* happens and if the *Condition* is met. During the transition, the *Action* is performed. Transitions between states can be caused either by commands from the MAC layer (MAC Request), by timers internal to the radio layer or by the start or the end of message reception on the radio medium. In the RADIO_DOZE state, the radio cannot send, nor receive. It consumes very little energy. On request from the MAC layer, the radio layer goes into the RADIO_READY_TX or RADIO_READY_RX states, after a waiting delay that depends from the originating state. The designed model is applicable to transceivers whose power average consumption is the same when setting up or turning around into the RADIO_RECEIVE or the RADIO_TRANSMIT state. This assumption was made for the WiseNET SoC at the beginning of the project. Measurements of the power consumption during transitions of a XE1203 transceiver has shown that a better model would differentiate the setup and turn-around states.

The transition to the RADIO_TRANSMIT state from the RADIO_READY_TX state is completely controlled by the MAC layer. The transition from the RADIO_READY_RX to the RADIO_RECEIVE state is triggered by the beginning of the reception of a message on the radio medium. For the message to be locked with success, it must be received with a power above the receiver's sensitivity, and large enough to present a given signal to noise ratio. A message is received with success if it presents the wanted signal to noise ratio during its whole duration, as illustrated in Fig. B.2. The algorithm to compute the accumulated noise and the capture behavior are illustrated in the finite state machine shown in Fig. B.3. This algorithm was not modified from the one in the original Glomosim accumulated noise radio model: A packet is

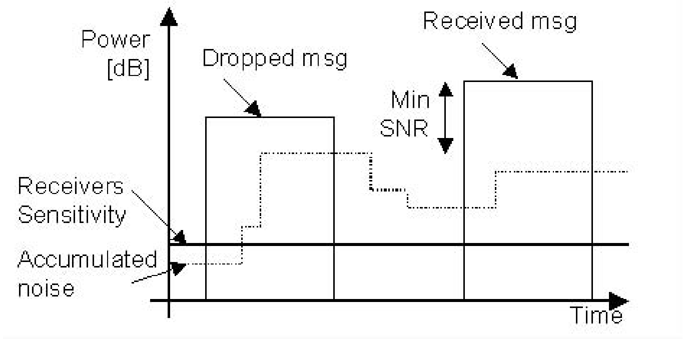

Figure B.2: Accumulated noise.

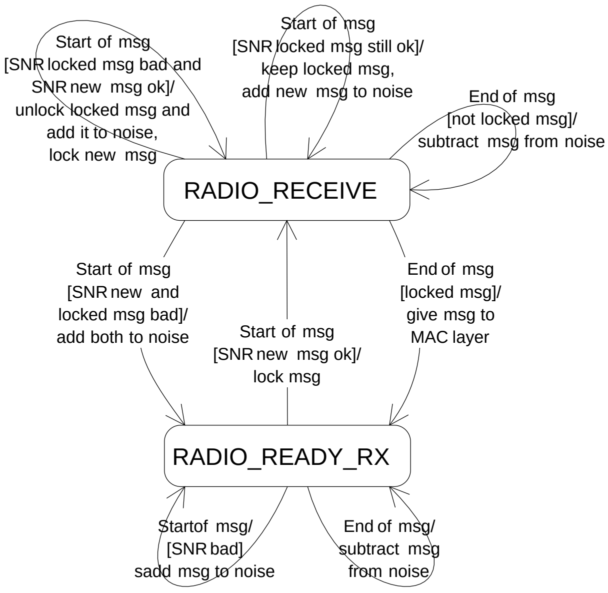
Start of msg [SNR locked msg bad and SNR new msg ok]/ unlock locked msg and add it to noise, lock new msg
Start of msg [SNR locked msg still ok]/ keep locked msg, add new msg to noise
End of msg [not locked msg]/ subtract msg from noise
RADIO_RECEIVE
Start of msg [SNR new and locked msg bad]/ add both to noise
Start of msg [SNR new msg ok]/ lock msg
End of msg [locked msg]/ give msg to MAC layer
RADIO_READY_RX
Startof msg/ [SNR bad] sadd msg to noise
End of msg/ subtract msg from noise

Figure B.3: Accumulated noise simulation model.

transmitted to all neighbors, with a different power attenuation and time delay as a function of the distance. At the receiving side, in the RADIO_READY_RX state, the new message is either added to the accumulated noise (if it is below the receiver sensitivity or if the SNR is below the threshold) or locked. Once a message is locked, the radio goes in the RADIO_RECEIVE state and continues to compute the noise curve, to check whether the SNR is still large enough. If the power of a new message reduces the SNR of the locked message below the wanted threshold, the locked message is dropped and its receive power is added to the noise. If the power of the new message is above the needed SNR, it is captured. Otherwise, it is added to the noise as well, and the radio goes into the RADIO_READY_RX state. Whenever the "End of msg" event is received, the message power is subtracted from the accumulated noise. If the message is still locked when the "End of msg" event is received, it is considered received without interferences and given to the MAC layer. In other states than RADIO_READY_RX and RADIO_RECEIVE, the "Start of msg" event only implies adding the message power to the accumulated noise. This message is never locked, whatever the SNR.

Bibliography

[1] N. Abramson. The ALOHA system - another alternative for computer communications. In *Proc. Fall Joint Computer Conf.*, pages 281–285, 1970.

[2] Norman Abramson. Development of the alohanet. *IEEE Transactions on Information Theory*, 31(2):119–123, March 1985.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *IEEE Communications Magazine*, 40(8):102–114, Aug. 2002.

[4] ZigBee Alliance. Network-specification, revision 5. Draft Version 0.80, 2003.

[5] Rimon Barr, Zygmunt J. Haas, and Robbert van Renesse. JiST: Embedding Simulation Time into a Virtual Machine. In *Proc. EuroSim Congress on Modelling and Simulation*, Sept. 2004.

[6] F. Bennett et al. Piconet: embedded mobile networking. *IEEE Journal on Personal Communications*, 4(5):8–15, Oct. 1997.

[7] D. Bertozzi, A. Raghunathan, S. Ravi, and L. Benini. Transport protocol optimization for energy efficient wireless embedded systems. In *Proc. ACM/IEEE Design, Automation, and Test in Europe (DATE)*, March 2003.

[8] Sanjay K. Bose. *An introduction to queueing systems.* Kluwer Academic / Plenum Publishers, 2002.

[9] J. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, 25(5):505 –515, Sept. 1979.

[10] A. Capone, M. Gerla, and R R. Kapoor. Efficient polling schemes for Bluetooth picocells. In *Proc. IEEE Int. Conf. on Communications*, pages 1990–1994, June 2001.

[11] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proc. 7th ACM Int. Conf. on Mobile Computing and Networking*, pages 85–96, Rome, Italy, July 2001.

[12] J.-C. Chen, K. M. Sivalingam, and P. Agrawal. Performance Comparison of Battery Power Consumption in Wireless Multiple Access Protocols. *Wireless Networks*, 5:445–460, 1999.

[13] P. Chen, B. O'Dea, and E. Callaway. Energy efficient system design with optimum transmission range for wireless ad hoc networks. In *Proc. IEEE Int. Conf. on Communications (ICC)*, pages 945–952, Apr. 2002.

[14] S.C.Q. Chen and V. Thomas. Optimization of inductive rfid technology. In *Proc. 2001 IEEE International Symposium on Electronics and the Environment*, pages 82 – 87, May 2001.

[15] Chipcon. CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. rev 1.2, 2004.

[16] C. Y. Chong, S. Mori, and C. C. Kuo. Information Fusion in Distributed Sensor Networks. In *Proc. of the IEEE American Control Conference*, volume 2, pages 830–835, New York, USA, 1985.

[17] Neiyer Correal and Neal Patwari. Wireless Sensor Networks: Challenges and Opportunities. In *Proc. Virginia Tech MPRG Symposium on Wireless Personal Communication*, Blacksburg, USA, June 2001.

[18] Eraldo Damosso, editor. *COST 231: Final Report - Digital Mobile Radio Towards Future Generation Systems.* COST Telecom Secretariat, 1997.

[19] Jing Deng and Zygmunt J. Hass. Dual Busy Tone Multiple Access (DBTMA): A New Channel Access Control for Packet Radio Networks. In *Proc. IEEE International Conference on Universal Personal Communications (ICUPC '98)*, volume 2, pages 973–977, Oct. 1998.

[20] E. W. Dijkstra. A Note on Two Problems in connection with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[21] Duracell. Alkaline-manganese dioxide. Technical Bulletin.

[22] Duracell. Lithium-manganese dioxide. Technical Bulletin.

[23] Jean-Pierre Ebert and Adam Wolisz. Power Saving in Wireless LANs: Analyzing the RF Transmission Power and MAC Retransmission Trade-Off. In *Proc. European Wireless*, 1999.

[24] Amre El-Hoiydi. Interference Between Bluetooth Networks - Upper Bound on the Packet Error Rate. *IEEE Communications Letters*, 5(6):245–247, June 2001.

[25] Amre El-Hoiydi. Packet Error Rate due to Interference Between Bluetooth Networks - Probabilistic Upper Bound and Simulation Results. In *Proc. Virginia Tech MPRG Symposium on Wireless Personal Communication*, pages 23–30, Blacksburg, USA, June 2001.

[26] Amre El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks. In *Proc. IEEE Int. Conf. on Communications*, pages 3418–3423, New York, USA, April 2002.

[27] Amre El-Hoiydi. Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks. In *Proc. IEEE Int. Conf. on Computers and Communications (ISCC)*, pages 685–692, Taormina, Italy, July 2002.

[28] Amre El-Hoiydi and Jean-Dominique Decotignie. Soft Deadline Bounds for Two-Way Transactions in Bluetooth Piconets under co-channel Interference. In *Proc. IEEE Int.*

Conf. on Emerging Technologies and Factory Automation (ETFA), pages 144–151, Antibes, France, Oct. 2001.

[29] Amre El-Hoiydi and Jean-Dominique Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. In *Proc. First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Lecture Notes in Computer Science, LNCS 3121*, pages 18–31. Springer-Verlag, July 2004.

[30] Amre El-Hoiydi, Jean-Dominique Decotignie, Christian Enz, and Erwan Le Roux. Poster Abstract: WiseMAC, An Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network. In *Proc. First ACM International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pages 302–303, Nov. 2003.

[31] EnOcean. Mesh networks. http://www.enocean.com, 2005.

[32] Christian Enz, Amre El-Hoiydi, Jean-Dominique Decotignie, and Vincent Pereis. Wisenet: An ultralow-power wireless sensor network solution. *IEEE Computer*, 37(8):62–70, Aug. 2004.

[33] ETSI. Paging Systems (PS); European Radio Message System (ERMES); Part 4: Air interface specification. ETS 300 133-4, July 1992.

[34] ETSI. HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1: Functional specification. ETS 300 652, Oct. 1996.

[35] ETSI. Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) layer; Part 2: Radio Link Control (RLC) sublayer. ETSI TS 101 761-2 V1.1.1, April 2000.

[36] Eveready Battery Co. E91 Energizer AA Alkaline. Engineering datasheet, Form No. EBC - 1102J, March 2005.

[37] Laura Marie Feeney and Martin Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In *Proc. IEEE INFOCOM*, pages 1548–1557, April 2001.

[38] International Organization for Standardization. Open Systems Interconnection - Basic Reference Model. ISO 7498:1984, 1984.

[39] E. N. Gilbert. Random Plane Networks. *Journal of the Society of Industrial and Applied Mathematics*, 9(4):533–543, Dec. 1961.

[40] Gray Girling, Jennifer Li Kam Wa, Paul Osborn, and Radina Stefanova. The PEN Low Power Protocol Stack. In *Proc. 9th IEEE International Conference on Computer Communications and Networks*, Las Vegas, Oct. 2000.

[41] Andrea J. Goldsmith and Stephen B. Wicker. Design challenges for energy-constrained Ad Hoc wireless networks. *IEEE Wireless Communications*, 9(4):8–27, Aug. 2002.

[42] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurithi. Packet Reservation Multiple Access for Local Wireless Communications. *IEEE Transactions on Communications*, 37(8):885–890, Aug. 1989.

[43] Ajay Chandra V. Gummalla and John O. Limb. Wireless Medium Access Control Protocols. *IEEE Communications Surveys and Tutorials*, pages 2–15, Second Quarter 2000.

[44] Paul J.M. Havinga. *Mobile Multimedia Systems.* PhD thesis, University of Twente, Feb. 2000.

[45] Wendi Rabiner Heinzelmann, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. Hawaii International Conference on System Sciences*, Jan. 2000.

[46] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. ACM/IEEE Int. conf. on Mobile Computing and Networking (MOBICOM'01)*, Rome, Italy, July 2001.

[47] Lifei Huang and Ten-Hwang Lai. On the Scalability of IEEE 802.11 Ad Hoc Networks. In *Proc. ACM MOBIHOC*, Lausanne, Switzerland, June 2000.

[48] Antenna Factor Inc. Data Guide SPLATCH Planar Antennas, Sept. 2004.

[49] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. Int. Conf. on Mobile Computing and Networking (MobiCOM)*, Aug. 2000.

[50] ITU-R. M.584-2 Codes and formats for radio paging, Nov. 1997.

[51] David B. Johnson, David A. Maltz, and Josh Broch. *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*, pages 139–172. In *Ad Hoc Networking*, edited by Charles E. Perkins Addison-Wesley, 2001.

[52] Edgar H. Callaway Jr. *Wireless Sensor Networks.* Auerbach CRC Press LCC, 2004.

[53] Phil Karn. MACA - A New Channel Access Method for Packet Radio. In *Proc. ARRL 9th Computer Networking Conf.*, Sept. 1990.

[54] M. J. Karol, Liu Zhao, and K. Y. Eng. Distributed-queueing request update multiple access (dqruma) for wireless packet (atm) networks. In *Proc. IEEE International Conference on Communications*, volume 2, pages 1224 – 1231, Seattle, USA, June 1995.

[55] O. Kasten and M. Langheinrich. First experiences with bluetooth in the smart-its distributed sensor network. In *Proc. Workshop on Ubiquitous Computing and Communications (PACT)*, Barcelona, Spain, Sept. 2001.

[56] J. Khun-Jush, P. Schramm, G. Malmgren, and J. Torsner. HiperLAN2: broadband wireless communications at 5 GHz. *IEEE Communications Magazine*, 40(6):130–136, June 2002.

[57] H. A. Kiehne, editor. *Battery Technology Handbook, Second Edition.* Marcel Dekker, Inc., 2003.

[58] Leonard Kleinrock and Fouad A. Tobagi. Packet Switching in Radio Channels: Part 1-Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, 1975.

[59] Ronny Krashinsky and Hari Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. In *Proc. 8th annual international conference on Mobile computing and networking (Mobicom)*, pages 119 – 130, Atlanta, Georgia, USA, Sept. 2002.

[60] K. Langendoen and G. Halkes. *Energy-Efficient Medium Access Control, Embedded Systems Handbook.* CRC press, 2005.

[61] Erwan Le Roux and Thierry Melly. Wisenet. CSEM Technical Report, Vers. 1.0, No 979, CSEM, Dec. 2001.

[62] Martin Leopold, Mads Bondo Dydensborg, and Philippe Bonnet. Bluetooth and sensor networks: A reality check. In *Proc. First ACM International Conference on Embedded Networked Sensor Systems (SenSys'03)*, Nov. 2003.

[63] Paul Lettieri, Christina Fragouli, and Mani B. Srivastava. Low Power Error Control for Wireless Links. In *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 139–150, Budapest, Hungary, Sept. 1997.

[64] J. Li et al. Capacity of Ad Hoc wireless networks. In *Proc. ACM/IEEE MOBICOM Conf*, pages 61–69, 2001.

[65] Jing Li and Georgios Y. Lazarou. A Bit-Map-Assisted Energy-Efficient MAC Scheme for Wireless Sensor Networks. In *Proc. Third International Symposium on Information Processing in Sensor Networks*, April 2004.

[66] Ting-Yu Lin and Ye-Chee Tseng. Collision analysis for a multi-bluetooth picocells environment. *IEEE Communication Letters*, 7(10):475–477, Oct. 2003.

[67] M Lott and I Forkel. A multi-wall-and-floor model for indoor radio propagation. In *Proc. Vehicular Technology Conference*, volume 1, pages 464 – 468, May 2001.

[68] Gang Lu, Bhaskar Krishnamachari, and Cauligi Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks. In *Proc. 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN 04), held in conjunction with the IEEE IPDPS Conference 18th International Parallel and Distributed Processing Symposium*, April 2004.

[69] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proc. Int. Workshop on Wireless Sensor Networks and Applications (WSNA)*, Sept. 2002.

[70] B. Mangione-Smith. Low power communications protocols: paging and beyond. In *Proc. IEEE Symposium on Low Power Electronics*, pages 8–11, Oct. 1995.

[71] Mark Weiser, Brent Welch, Alan Demers, Scott Shenker. Scheduling for Reduced CPU Energy. In *Proceedings of the First Symposium on Operating Systems Design and Implementation*, pages 13–23. Usenix Association, Nov. 1994.

[72] Franco Mazzenga, Dajana Cassioli, Pierpaolo Loreti, and Francesco Vatalaro. Evaluation of packet loss probability in bluetooth networks. In *Proc. International Conference on Communications*, pages 313–317, New-York, USA, April 2002.

[73] M. Metcalf and D. R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM*, 19(9):395–404, July 1976.

[74] MicroStrain. Wireless web sensor networks (wwsn). http://www.microstrain.com, 2005.

[75] Rex Min and Anantha Chandrakasan. A Framework for Energy-Scalable Communication in High-Density Wireless Networks. In *Proc. International Symposium on Low Power Electronics and Design, ISLPED'02*, pages 36–41, Monterey, California, USA, Aug. 2002.

[76] M Molle and L Kleinrock. Virtual time csma: Why two clocks are better than one. *IEEE Transactions on Communications*, 33(9):919–933, Sep 1985.

[77] Michel Mouly and Marie-Bernadette Pautet. *The GSM System for Mobile Communications.* Telecom Publishing, 1992.

[78] Randolph Nelson and Leonard Kleinrock. Spatial TDMA: A Collision-Free Multihop Channel Access Protocol. *IEEE Transactions on Communications*, 33(9):934–944, Sept. 1985.

[79] LAN MAN Standards Committee of the IEEE Computer Society. 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE, 1999.

[80] LAN MAN Standards Committee of the IEEE Computer Society. 802.5 - Token Ring Access Method and Physical Layer Specification. IEEE 802.5, 1998 Edition (ISO/IEC 8802-5:1998), 1999.

[81] LAN MAN Standards Committee of the IEEE Computer Society. 802.3 - Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. IEEE, 2002.

[82] LAN MAN Standards Committee of the IEEE Computer Society. IEEE 802.15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANS). IEEE, 2003.

[83] OPNET Technologies, Inc. web site. http://www.opnet.com/.

[84] Guangyu Pei and Charles Chien. Low Power TDMA in Large Wireless Sensor Networks. In *Proc. IEEE Military Communications Conference MILCOM*, volume 1, pages 347–351, Oct. 2001.

[85] V. Peiris, E. Leroux, T. Melly, F. Pengg, D. Ruffieux, N. Raemy, F. Giroud, A. Ribordy, M. Kucera, R. Caseiro, C. Arm, S. Cserveny, P.-D. Pfister, P. Volet, and S. Gyger. Implementation of the WiseNET Low-Power 1V RF CMOS Transceiver IC for Wireless Sensor Networks. *CSEM Scientifique and Technical Report*, 2004.

[86] Brian S. Peterson, Rusty O. Baldwin, Jeffrey P. Kharoufeh, and Richard A. Raines. Refinements to the packet error rate upper bound for Bluetooth networks. *IEEE Communications Letters*, 7(8):382–384, Aug. 2003.

[87] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proc. Second ACM International Conference on Embedded Networked Sensor Systems (SenSys'04)*, Nov. 2004.

[88] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[89] J. Rabaey et al. Picoradio supports ad hoc ultra-low power wireless networking. *IEEE Computer Magazine*, pages 42–48, July 2000.

[90] Jonathan M. Reason and Jan M. Rabaey. A study of energy consumption and reliability in a multi-hop sensor network. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(1):84–97, 2004.

[91] Sokwoo Rhee, Deva Seetharam, Shen Liu, Ningya Wang, and Jason Xiao. i-beans: An ultra-low power wireless sensor network. In *Proc. Ubicomp 2003, The Fifth International Conference on Ubiquitous Computing*, Seattle, WA, Oct. 2003.

[92] Matthias Ringwald and Kay Römer. BitMAC: A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks. In *Proc. 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, pages 57–69, Jan. 2005.

[93] Raphael Rom and Moshe Sidi. *Multiple Access Protocols - Performance and Analysis.* Springer-Verlag, 1990.

[94] A. K. Salkintzis and C. Chamzas. Performance analysis of a downlink MAC protocol with power-saving support. *IEEE Transactions on Vehicular Technology*, 49(3):1029–1040, May 2000.

[95] A. K. Salkinzis and C. Chamzas. Mobile packet data technology: an insight into MOBITEX architecture. *IEEE Personal Communications*, 4(1):10–18, Feb. 1997.

[96] SaRonix. 32.768 kHz Tubular Crystal NTF3238 / NTF3226 Series. Datasheet, DS-118 REV D.

[97] Scalable Network Technologies, Inc. http://www.scalable-networks.com/.

[98] Curt Schurgers, Olivier Aberthorne, and Mani B. Srivastava. Modulation Scaling for Energy Aware Communication Systems. In *Proc. International Symposium on Low Power Electronics and Design, ISLPED'01*, pages 96–99, Juntington Beach, California, USA, Aug. 2001.

[99] Curt Schurgers, Vlasios Tsiatsis, and Mani B. Srivastava. STEM: Topology Management for Energy Efficient Sensor Networks. In *Proc. IEEE Aerospace Conf.*, volume 3, pages 1099–1108, March 2002.

[100] Scott Y. Seidel and Theodore S. Rappaport. 914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings. *IEEE Transcactions on Antennas and Propagation*, 40(2):207–217, Feb. 1992.

[101] E. Shih, P. Bahl, and M. J. Sinclair. Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *Proc. Eighth Annual ACM Conf. on Mobile Computing and Networking*, pages 160–171, Altanta, Georgia, USA, Sept. 2002.

[102] Bluetooth SIG. Specification of the Bluetooth System - Core - Version 1.1, Feb. 2001.

[103] S. Singh and C. Raghavendra. PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks. *ACM SIGCOMM Computer Communication Review*, 28(3), July 1998.

[104] K. M. Sivalingam, Mani Srivastava, Prathima Agrawal, and Jyh-Cheng Chen. Low-power access protocols based on scheduling for wireless and mobile atm networks. In *Proc. IEEE International Conference on Universal Personal Communications (ICUPC)*, pages 429–433, San Diego, CA, USA, Oct. 1997.

[105] Bernard Sklar. *Digital Communications, Fundamentals and Applications.* Prentice-Hall, 1988.

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

[107] John A. Stine and Gustavo De Veciana. Improving energy efficiency of centrally controlled wireless data networks. *Wireless Networks*, 8(6):681–700, Nov. 2002.

[108] Sameer Sundresh, Kim WooYoung, and Agha Gul. SENS: A Sensor, Environment and Network Simulator. In *Proc. 37th Annual Simulation Symposium (ANSS37)*, Arlington, VA, USA, April 2004.

[109] Robert Szewczyk, Eric Osterweil, Joseph Polastre, Michael Hamilton, Alan Mainwaring, and Deborah Estrin. Habitat Monitiring with Sensor Networks. *Communications of the ACM*, 34-40(6):24, June 2004.

[110] H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, 32(3):246 –257, March 1984.

[111] Andrew S. Tanenbaum. *Computer Networks.* Prentice-Hall, second edition, 1989.

[112] Lucent Technologies. Orinoco Silver PC Card - Getting Started Guide, Aug. 2000.

[113] The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[114] Fouad A. Togabi and Leonard Kleinrock. Packet Switch in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, Dec. 1975.

[115] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. In *Proc. IEEE INFOCOM*, pages 200–209, New York, June 2002.

[116] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. First ACM International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pages 171–180, Nov. 2003.

[117] L. F. W. van Hoesel, T. Nieberg, H. J. Kip, and P. J. M. Havinga. Advantages of a TDMA based, energy-efficient, self-organizing mac protocol for WSNs. In *Proc. IEEE 59th Vehicular Technology Conference, VTC 2004-Spring*, volume 3, pages 1598–1602, May 2004.

[118] L.F.W. van Hoesel and P.J.M. Havinga. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches. In *Proc. INSS*, Japan, June 2004.

[119] John von Neumann. First draft of a report on the edvac. Technical report, University of Pennsylvania, June 1945.

[120] Geoffrey Werner-Allen, Jeff Johnson, Mario Ruiz, Jonathan Lees, and Matt Welsh. Monitoring Volcanic Eruptions with a Wireless Sensor Network. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN'05)*, Jan. 2005.

[121] H. Woesner, J.-P. Ebert, M. Schläger, and A. Wolisz. Power saving mechanisms in emerging standards for wireless lans: the mac level perspective. *IEEE Personal Communications*, pages 40–48, June 1998.

[122] Alec Woo and David Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proc. ACM Mobicom*, Rome, Italy, July 2001.

[123] Xemics. CoolRISC 816 8-bit Microprocessor Core - Hardware and Software Reference Manual. Version 4.5, Apr. 2001.

[124] Xemics. XE88LC06A Ultra Low-Power Low-Voltage Radio Machine. Datasheet, Aug. 2002.

[125] Xemics. XE8000 driving XE1200 transceivers standard API. Software, V1.0, March 2003.

[126] Xemics. XM1203 868/915 MHz Transceiver Board. Product Brief, 2003.

[127] Xemics. XE1203 TrueRF 433 MHz/868MHz/915MHz Low-Power, Integrated UHF Transceiver. Datasheet, 2004.

[128] Xemics. XE800EV108 Evaluation Board for the XE88LC06AMIO26 and the XE88LC07AMIO26. User's Guide, 2005.

[129] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A Wireless Sensor Network For Structural Monitoring. In *Proc. ACM SenSys*, Baltimore, Maryland, USA, Nov. 2004.

[130] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed Energy Conservation for Ad hoc Routing. In *Proc. ACM/IEEE Mobicom Conf.*, pages 70–84, 2001.

[131] Wei Ye, John Heidemann, and Deborah Estrin. A Flexible and Reliable Radio Communication Stack on Motes. Technical report ISI-TR-565, USC/ISI, Sept. 2002.

[132] Wei Ye, John Heidemann, and Deborah Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. IEEE INFOCOM Conf.*, 2002.

[133] Wei Ye, John Heidemann, and Deborah Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. Technical Report ISI-TR-567, USC/Information Sciences Institute, Jan. 2003.

[134] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. In *Proc. Int. Workshop on Parallel and Distributed Simulations*, pages 154–161, May 1998.

[135] Michele Zorzi and Ramesh R. Rao. Energy Constrained Error Control for Wireless Channels. *IEEE Personal Communications*, pages 27–33, Dec. 1997.

[136] S Zuerbes. Considerations on Link and System Throughput of Bluetooth Networks. In *Proc. IEEE Int. Symp. on Personal, Indoor, and Mobile Communications, PIMRC*, pages 1315–1319, London, Sept. 2000.

[137] S. Zuerbes, W. Stahl, K. Matheus, and J Haartsen. Radio Network Performance of Bluetooth. In *Proc. IEEE Int. Conf. on Communications*, pages 1563–1567, New Orleans, June 2000.

Curriculum Vitae

Amre El-Hoiydi

Born in Brussels in 1971. Swiss citizen.

Education

2001 - 2005	Docteur ès Sciences (PhD), School of computer & communication sciences, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland.
2000 - 2001	Graduate School in Communications Systems, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland.
1989 - 1994	Master of Science (MSc) in electrical engineering, Swiss Federal Institute of Technology Zurich, (ETHZ), Switzerland. (Thesis done at the Imperial College, London, UK.)
1986 - 1989	High school in Porrentruy, Jura, Switzerland.
1977 - 1986	Primary and secondary school in Delémont, Jura, Switzerland.

Professional Experience

1996 - now (9 years)	Swiss Center for Electronics and Microtechnology (CSEM). R&D Engineer. ▷ *Research in the field of low power communication protocols for wireless sensor networks (CSEM WiseNET and NCCR MICS research projects).* ▷ *Electronic design and embedded programming of VxWorks and Linux based wireless LAN gateways (European ESPRIT INFOGATE and IST OPEN-ROUTER research projects).* ▷ *Contributions to the specification and implementation of a network demonstrator of the Universal Mobile Telecommunications System (UMTS) (European ACTS RAINBOW research project).* ▷ *Contributions to the research effort on the UMTS air interface (European ACTS FRAMES research project).*
1995-1996 (1.5 years)	Swiss Federal Institute of Technology in Lausanne (EPFL), Telecommunication laboratory. Teaching assistant. ▷ *Contributions to mobility management for low earth orbit mobile satellite communication systems, and their integration in the network architecture of UMTS (European RACE MONET research project).*

Publications

A. El-Hoiydi and J.-D. Decotignie. Low Power Mac Protocols for Infrastructure Wireless Sensor Networks. *ACM Mobile Networks and Applications (MONET), Special Issue on "WLAN Optimization at the MAC and Network Levels"*, accepted for publication.

C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Pereis. Wisenet: An Ultra Low Power Wireless Sensor Network Solution. *IEEE Computer*, 37(8):62–70, August 2004.

A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks. In *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Lecture Notes in Computer Science, LNCS 3121*, pages 18–31. Springer-Verlag, July 2004.

A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. In *Proceedings of the International Symposium on Computers and Communications ISCC*, pages 244–251, Alexandria, Egypt, June 2004.

A. El-Hoiydi, J.-D. Decotignie, and Jean Hernandez. Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. In *Proc. European Wireless (EW'04)*, pages 563–569, Barcelona, Spain, February 2004.

A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux. Poster Abstract: WiseMAC, An Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Network. In *Proc. 1st ACM SenSys Conf.*, pages 302–303, November 2003.

A. El-Hoiydi and J.-D. Decotignie. Simulation of Low Power MAC Protocols for Wireless Sensor Networks. In *Proc. IFAC Int. Conf. on Fieldbus Systems and their Applications*, pages 65–72, Aveiro, Portugal, July 2003.

A. El-Hoiydi. Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks. In *Proc. IEEE Int. Conf. on Computers and Communications (ISCC)*, pages 685–692, Taormina, Italy, July 2002.

A. El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks. In *Proc. IEEE Int. Conf. on Communications*, pages 3418–3423, New York, USA, April 2002.

J.-D. Decotignie, P. Dallemagne, and A. El-Hoiydi. Architectures for the Interconnection of Wireless and Wireline Fieldbusses. In *Proc. 4th IFAC International Conference on Fieldbus Systems and their Applications (FET)*, pages 285–290, Nancy, France, Nov. 2001.

A . El-Hoiydi and J.-D. Decotignie. Soft Deadline Bounds for Two-Way Transactions in Bluetooth Piconets under co-channel Interference. In *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 144–151, Antibes, France, October 2001.

A. El-Hoiydi. Packet Error Rate due to Interference Between Bluetooth Networks - Probabilistic Upper Bound and Simulation Results. In *Virginia Tech MPRG Symposium on Wireless Personal Communication*, pages 23–30, Blacksburg, USA, June 2001.

A. El-Hoiydi. Interference Between Bluetooth Networks - Upper Bound on the Packet Error Rate. *IEEE Communications Letters*, 5(6):245–247, June 2001.

A. El-Hoiydi. Implementation Options for the Distribution System in the 802.11 Wireless LAN Infrastructure Network. In *Proc. IEEE Int. Conf. on Communications (ICC)*, volume 4, pages 164 –169, New Orleans, USA, June 2000.

A. El-Hoiydi. Radio Independence in the Network Architecture of the Universal Mobile Telecommunication System. In *Proc. IEEE Global Telecommunications Conf. (Globecom)*, volume 4, pages 1957–1962, Sydney, Australia, November 1998.

G. Fleming, A. El-Hoiydi, J. De Vriendt, G. Nikolaidis, F. Piolini, and M. Maraki. A Flexible Network Architecture for UMTS. *IEEE Personal Communications Magazine*, 5(2):8–15, April 1998.

A. El-Hoiydi. Network Architecture of the UMTS Space Segment. In *Proc. 4th European Conf. on Satellite Communications (ECSC)*, pages 254–259, Rome, Italy, November 1997.

A. Saidi, G. Fleming, M. Barry, A. El-Hoiydi, B. Perrin, G. Nikolaidis, I. Modeas, and F. Piolini. RAINBOW Demonstrator Transport Chain. In *Proc. ACTS Mobile Communication Summit*, volume 1, pages 297–302, Aalborg, Denmark, October 1997.

G. Fleming, G. Nikolaidis, L. von Allmen, N. Alonistioti, M. Maraki, A. El-Hoiydi, and B. Perrin. Architecture and Design of the Rainbow Mobile Terminals, Base Stations and Mobility Servers. In *Proc. ACTS Mobile Communication Summit*, volume 1, pages 291–296, Aalborg, Denmark, October 1997.

A. El-Hoiydi and R. J. Finean. Location Management for the Satellite - Universal Mobile Telecommunication System. In *Proc. IEEE Int. Conf. on Universal Personal Communications (ICUPC)*, volume 2, pages 739 –744, Boston, MA, USA, September 1996.

J. P. Castro and A. El-Hoiydi. Space Segment Integration in Future Mobile Systems. In *Proc. Int. Mobile Satellite Conf. (IMSC)*, pages 433–438, Pasadena, CA, USA, June 1997.

R. J. Finean and A. El-Hoiydi. Satellite UMTS Network Architecture. In *Proc. IEE Conf. on Satellite Systems for Mobile Communications and Navigation*, pages 151–155, May 1996.

R. J. Finean, D Polymeros, A El-Hoiydi, F da Costa, M Dinis, A Saïdi, and B Vazvan. Impact of Satellites on UMTS Network. In *Proc. RACE Mobile Telecommunications Summit*, pages 388–392, Cascais, Portugal, November 1995.