

Power-Aware Adaptive Techniques for Wireless Sensor Networks

Muhammad Mahtab Alam

▶ **To cite this version:**

Muhammad Mahtab Alam. Power-Aware Adaptive Techniques for Wireless Sensor Networks. Networking and Internet Architecture [cs.NI]. Université Rennes 1, 2013. English. NNT : . tel-00931860

HAL Id: tel-00931860

https://theses.hal.science/tel-00931860

Submitted on 16 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : ANNÉE 2013

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour la grade de
DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention : Traitement du Signal et Télécommunications
École Doctorale : MATISSE
présentée par

Muhammad Mahtab ALAM

préparée à l'unité de recherche : IRISA - UMR6074
Institut de Recherche en Informatique et Systèmes Aléatoires - CAIRN
École Nationale Supérieure des Sciences Appliquées et de Technologie

---

Techniques adaptatives pour la gestion de l'energie dans les réseaux capteurs sans fil

Power-Aware Adaptive Techniques for Wireless Sensor Networks

Thèse soutenue à Lannion
le 26 février 2013

devant le jury composé de :

Tanguy RISSET
Professeur
INSA Lyon/ Rapporteur

Emanuel POPOVICI
Professeur
University College Cork / Rapporteur

Alain PEGATOQUET
Maitre de Conférences, Université de Nice, LEAT / Examiner

Daniel MÉNARD
Professeur
INSA Rennes / Examinateur

Olivier SENTIEYS
Professeur
Université de Rennes 1 / Directeur de thèse

Olivier BERDER
Maitre de Conférences (HDR)
Université de Rennes 1 / Co-directeur de thèse

Abstract

More and more nomadic devices embed both control and processing capabilities and one or more radio transceivers. These items include mainly wireless sensor networks (WSNs) or more generally the domain of wireless communicating objects. Their common point is that they must operate as long as possible without replacing or recharging batteries. Therefore, energy consumption is the main constraint to be considered in the development of these devices. Research on energy optimization are numerous, and are applied at all levels of design.

In this thesis, we propose algorithmic-level optimization techniques for energy reduction. In this context, first we proposed a pragmatic and precise hybrid energy model for WSN. This model takes into account different scenarios that occur during the communication and evaluates their energy consumption based on the software as well as the hardware components. The model presented is a combination of analytical approach and real time measurements. The validation of the model shows that the relative error is between 1 to 8 percent.

In the second part of the thesis we focused on the medium access control (MAC) layer. MAC layer plays a pivotal role for energy management in WSN because the radio transceiver (most energy consuming component) activity is being controlled by MAC layer. Moreover, idle listening is the dominant energy waste in most of the MAC protocols. In this regard, we propose a novel energy efficient traffic-aware dynamic MAC protocol (TAD-MAC). The protocol relies on dynamic adaptation of wake-up interval based on a traffic estimation which helps to efficiently keep the track of varying traffic. A heuristic-based approach is used to model the system by characterizing each parameter of the adaptive algorithm that is used for dynamic adaptation of wake-up interval. A detailed analysis on the convergence and performance metric for reaching a steady state is presented and evaluated. Furthermore, TAD-MAC is applied in the context of wireless body area sensor networks for fixed and variable traffic rates. TAD-MAC outperforms other low power MAC protocols in terms of latency as well as energy consumption and

consequently increases the lifetime by 3 to 6 times in comparison with other low power MAC protocols.

In the final part of the thesis an adaptive transmit power optimization technique is applied under dynamic channel variations to reduce the energy per successfully transmitted bit at the physical layer. The output power is adaptively tuned to the best power level on link-by-link basis. Each node locally adapts the power according to the signal-to-noise ratio (SNR) variations (for all the neighbor nodes). The optimization is achieved under a slow varying channel at the reception of a single packet. Different radio transceivers power profiles are used to show the gain over non-adaptive fixed transmit power. It is found that by dynamically adapting the transmit power the energy consumption can be reduced by a factor of 2.

Résumé

De plus en plus d'objets électroniques nomades incorporent à la fois des capacités de traitement, de contrôle et un ou plusieurs émetteur-récepteur radios. Les objets, lorsqu' ils communiquent entre eux, forment des réseaux de capteurs sans fil (WSN). Leur point commun est qu'ils doivent fonctionner aussi longtemps que possible sans avoir à remplacer ou recharger les batteries. Par conséquent, la consommation d'énergie est la principale contrainte à prendre en considération dans l'élaboration de ces articles. Les recherches sur l'optimisation de l'énergie sont nombreuses, et sont appliquées à tous les niveaux de la conception. Dans cette thèse, nous proposons des techniques d'optimisation au niveau algorithmique visant une réduction de l'énergie consommée.

Dans ce contexte, tout d'abord, nous avons proposé un modèle énergétique hybride pragmatique et précis pour les WSN. Ce modèle prend en compte les différents scénarios qui se produisent pendant la communication et évalue leur consommation d'énergie en distinguant composantes logicielles et matérielles. Le modèle présenté est une combinaison d'approche analytique et de mesures en temps réel. La validation du modèle montre que l'erreur relative est de 1 à 8 pour cent.

Dans la deuxième partie de la thèse nous nous sommes concentrés sur le contrôle d'accès au medium (MAC). La couche MAC joue un rôle essentiel dans la gestion de l'énergie dans les réseaux de capteurs, car l'activité de l'émetteur-récepteur radio (composante la plus gourmande en énergie) est contrôlée par la couche MAC. Par ailleurs, l'état d'écoute " idle " se trouve être l'état dans lequel s'opère un gaspillage énorme d'énergie dans la plupart des protocoles MAC. À ce sujet, nous proposons un nouveau protocole MAC dynamique (TAD-MAC) efficace en énergie avec une connaissance à priori du trafic courant. Le protocole repose sur l'adaptation dynamique des intervalles de réveil basée sur une estimation du trafic. Une approche heuristique est utilisée pour modéliser le système en caractérisant chacun des paramètres de l'algorithme adaptatif. Une analyse détaillée de la convergence et des ls métriques de performance pour atteindre un état d'équilibre est présentée et évaluée. En outre, TAD-MAC est appliqué dans le contexte des réseaux corporels de capteurs sans fil pour des taux de trafics fixes et variables.

TAD-MAC surpasse les autres protocoles MAC à faible consommation énergétique en termes de latence ainsi que de consommation d'énergie et donc permet d'augmenter la durée de vie de trois à six fois.

Dans la dernière partie de la thèse une technique adaptative d'optimisation de la puissance d'émission est appliquée avec une variation dynamique des canaux de transmission afin de réduire l'énergie par bit transmis avec succès au niveau de la couche physique. La puissance d'émission est réglée de manière adaptative sur un critère de type " link-to-link " . Chaque noeud adapte localement sa puissance en fonction des variations du rapport signal-sur-bruit (SNR) (pour tous les noeud voisins). Différents profils d'émetteurs-récepteurs radio sont utilisés pour montrer le gain par rapport à l'utilisation d'une puissance d'émission fixe. Il se trouve que par l'adaptation dynamique de la puissance d'émission, la consommation d'énergie peut être réduite d'un facteur 2.

Acknowledgment

First of all, countless thanks to Almighty Allah, Who has given me an opportunity, knowledge and courage to carry out this work. Secondly, I would like the readers to keep all the below-mentioned people in mind while reading this dissertation, most definitely without them this success would not have been possible.

I would start by thanking my PhD supervisors, Olivier Sentieys, Olivier Berder and Daniel Menard for their guidance, end-less support, always have belief in my capabilities, understanding and patience. Their professionalism, experience, commitment and dedication has always inspired me to do well not only in research but also in daily professional life. They were not only my advisers, bosses and professional mentors but also like a family member. It is my honor to have worked with them. I would like to say a special thank to Olivier Berder, who has been extremely supportive in all matters of concerns during my PhD, he has been just like a brother to me.

After that, I must thank and acknowledge on one side the Conseil Général des Côtes d'Armor and on the other side The European Union through GEODES project for their fundings on my PhD thesis. I also thank Centre National de la Recherche Scientifique (CNRS) for their generous financial support to attend national conference (ECOFAC-10) and winter school (FETCH-12).

I would also like to thank all my colleagues of CAIRN research team and administration staff at ENSSAT for creating such a pleasant work environment and for being there for me. In particular many thanks to Romain, Arnaud Carer and Thomas Anger for their practical support from the Laboratory. I would like to thank Pascal Scalart and Arnaud Tisserand for their fruitful discussions. Many thanks to all my colleagues including Daniel Chillet, Sébastian Pillement, Emmanuel Casseau, Philippe Quemerais, Robin, Vivek, Shafqat, Karthick, Vinh, Nhan, Rheifeng and others. I would also like to pay special thanks to CAIRN team-assistants Joelle Thepault, Angelique and Nadia for their support. Also I would like to thank my French language teacher Nathelie Caradec at ENSSAT for helping me to learn basic French.

Most importantly I would like to thank my parents and family for their faith in me, encouraging me to be as ambitious as I wanted and supporting me in all my endeavors. It was the encouragement and prayers of my mom and dad which has elevate me to where I am today. A great thanks to my lovely wife without her motivations and support it would have might not possible. She had been my strength to achieve many tasks and goals during this endeavors. A very special thanks goes to my eldest brother and to all my other brothers and sisters, nephews and nieces for being there for me all the time and cheering me up. I thank all my friends in France, Denmark and in Pakistan for motivating me and regularly wishing me good luck.

Finally I would like to express my gratitude to all the jury members for spending their precious time to read and accept my work.

Last but not the least, I would like to dedicate this thesis to my sweet daughter Yasmine, who had always been a reason of motivations, encouragements and love.

Contents

List of Figures x

List of Tables xvi

Abbreviations xviii

Résumé étendu 1
- 0.1 Introduction . . . 1
- 0.2 Méthodologie d'optimisation . . . 3
- 0.3 Modèle de consommation d'énergie . . . 4
  - 0.3.1 Mesures expérimentales . . . 6
    - 0.3.1.1 Plate-forme PowWow . . . 6
    - 0.3.1.2 Couche MAC asynchrone . . . 8
  - 0.3.2 Expérimentations . . . 10
  - 0.3.3 Modèle hybride . . . 11
- 0.4 Adaptation dynamique de l'intervalle de réveil des capteurs . . . 14
- 0.5 Adaptation dynamique de la puissance d'émission . . . 20
- 0.6 Conclusions . . . 23

1 Introduction 25
- 1.1 Motivation and Objectives . . . 27
- 1.2 Contributions . . . 29
- 1.3 Thesis Organization . . . 31

2 Wireless Sensor Networks 33
- 2.1 Reduced Open Systems Interconnection (OSI) model . . . 34
- 2.2 Medium Access Control . . . 35
  - 2.2.1 Berkeley-MAC Protocol . . . 37
  - 2.2.2 Reduced Preamble XMAC Protocol . . . 38
  - 2.2.3 WiseMAC protocol . . . 39
  - 2.2.4 Cross-Layer Optimizations . . . 40
- 2.3 Energy Modeling . . . 42
  - 2.3.1 Parameter-based Power Model . . . 42
  - 2.3.2 State-based Power Model . . . 43
  - 2.3.3 Component-based Power Model . . . 44
- 2.4 Hardware Platforms and Network Simulators . . . 45
  - 2.4.1 Sensor Platforms . . . 45
  - 2.4.2 Network Simulators . . . 47

2.5 Conclusion . . . 51

3 Accurate Energy Consumption Modeling of WSNs 52
3.1 Related Work . . . 54
3.2 Partial Cross-Layer Modeling . . . 55
3.2.1 Data Link Layer Model . . . 55
3.2.2 Medium Access Control Model . . . 56
3.2.2.1 **Collision Probability Model** . . . 59
Wake-Up Collision . . . 60
Data Collision . . . 60
3.2.2.2 **Analytical Analysis of Power Consumption in RICER3** 60
3.2.3 Physical-Layer Model . . . 62
3.2.4 Channel Model . . . 64
3.2.5 A Brief Conclusion on the Analytical Model . . . 64
3.3 Protocol Implementation and Power Measurements . . . 65
3.3.1 Optimizations of MAC Parameters . . . 66
3.3.2 Platform Power Measurements . . . 67
3.4 Scenario-Based Hybrid Energy Model . . . 72
3.4.1 Scenario Descriptions and Cost Analyses . . . 74
3.4.1.1 Basic Cost . . . 76
3.4.1.2 Re-transmission Cost . . . 76
3.4.1.3 Data Collision Cost . . . 76
3.4.1.4 Wake-Up-Collision Cost . . . 77
3.4.1.5 Total Energy Cost . . . 77
3.4.2 Hybrid Energy Model . . . 77
3.4.3 Energy Consumption Measurements . . . 78
3.5 Validation and Performance Evaluation . . . 79
3.6 Applications of the Energy Model . . . 82
3.6.1 Effects of Collisions and Total Energy . . . 82
3.6.2 Optimization of the Output Transmit Power . . . 83
3.7 Conclusion . . . 83

4 TAD-MAC: Traffic-Aware Dynamic MAC Protocol for Wireless Sensor Networks 86
4.1 Introduction . . . 86
4.2 MAC Protocols Design Principles . . . 88
4.3 State of the Art . . . 89
4.4 Traffic-Aware Dynamic MAC Protocol . . . 91
4.4.1 Protocol Description . . . 91
4.4.2 Traffic Estimation . . . 93
4.4.3 Adaptive Algorithm . . . 96
4.5 Heuristic-Based Approach for System Characterization . . . 98
4.5.1 Experimental Setup of WSNet . . . 100
4.5.2 Open Loop System . . . 100
4.5.2.1 Behavior of μ in open-loop . . . 102
4.5.2.2 Behavior of e in open-loop . . . 102
4.5.3 Closed-Loop System . . . 104

4.5.3.1 Closed-loop form validation of μ . . . 105
4.5.3.2 Behavior of the combined μ and e in closed-loop . . . 106
4.5.3.3 A brief conclusion on closed-loop system . . . 108
4.6 Implementation And Performance Analysis . . . 109
4.6.1 Performance Analysis . . . 113
4.6.1.1 A brief conclusion on performance analysis of TAD-MAC 116
4.7 Conclusion . . . 117

5 Application of TAD-MAC Protocol for Wireless Body Area Sensor Networks 119
5.1 Introduction to WBASN Context . . . 119
5.2 Related Work . . . 122
5.3 Applying TAD-MAC for WBASN . . . 124
5.4 Performance Evaluation . . . 128
5.4.1 Energy Consumption Evaluation for Fixed Traffic . . . 128
5.4.2 Energy Consumption Evaluation for Variable Traffic . . . 132
5.5 Conclusion . . . 134

6 Adaptive Transmit Power Optimizations for WSN 135
6.1 Introduction . . . 135
6.2 Related Work . . . 137
6.2.1 Link Quality Estimators . . . 138
6.2.1.1 SNR-Based Link Quality Estimation . . . 138
6.2.1.2 Multiple Metric-Based Link Quality Estimation . . . 139
6.2.1.3 Chip Error-Based Link Quality Estimation . . . 139
6.2.2 Link Quality Estimation for Adaptive Transmit Power . . . 140
6.3 Transmit Power Optimization for WSN . . . 140
6.3.1 Power Profiles of Radio Transceivers . . . 140
6.3.2 Performance of Static Transmit Power . . . 142
6.3.3 Channel Profiles . . . 144
6.4 Adaptive Transmit Power Technique . . . 146
6.4.1 Algorithm . . . 147
6.4.2 Simulation Results . . . 149
6.5 Performance Evaluation . . . 153
6.6 Implementation Considerations . . . 156
6.7 Conclusion . . . 158

Conclusion and Perspectives 159

List of Figures

1 Méthodologie d'optimisation par exploration de paramètres 4
2 Méthodes d'évaluation de la consommation d'énergie 5
3 Architecture matérielle de la plate-forme PowWow 8
4 Photo de la plate-forme PowWow avec trois cartes: calcul, récupération d'énergie et radio 8
5 Les paquets de contrôle (WUB et ACK) comprennent l'adresse géographique du nœud qui les émettent, un code de redondance cyclique (CRC) et le dernier octet contient le type de paquet (3 bits) et le mode de communciation (5 bits). Le paquet de données contient les adresses de la source, de la destination, du dernier et du prochain nœud en cas de multi-étapes, ainsi que quelques octets pour le numéro d'ACK, les informations des capteurs, telle la température, le niveau de batterie et le numéro de trame. 9
6 Protocole RICER optimisé ajustable à l'application et diminution de l'écoute inutile. 9
7 Communication ratée pour cause de collision de données entre les deux émetteurs. 11
8 Phases de communication entre émetteurs et récepteurs. Le modèle d'énergie est basé sur les traces d'exécution des différents scénarios possibles durant la communication. 12
9 Modèle d'énergie hybride à partir de résultats expérimentaux et d'expressions analytiques. 13
10 Comparaison de notre modèle d'énergie avec différentes sessions de mesures pour 3 tests de $300s$ avec des valeurs différentes pour TWU et $WUInt$. . 13
11 Adaptation de l'intervalle de réveil aux messages des différents capteurs . 15
12 Algorithme de mise à jour de l'intervalle de réveil. TSR désigne le registre de statut du trafic, dans lequel '1' veut dire que le nœud a reçu un paquet qui lui était destiné, tandis que '0' révèle un réveil inutile. 16
13 Vue d'ensemble du système adaptatif. 17
14 Optimisation de la longueur du TSR et vitesse de convergence associée en fonction du trafic (de 2 paquets par seconde à 1 paquet toutes les 10 secondes). 18
15 Durée de vie d'un nœud (batterie alkaline AA) pour différents protocoles MAC et transceivers radio. Les transceivers $amis52100$ et $cc1000$ sont préférables pour de faibles trafics tandis que le $cc2420$ convient à des trafics plus élevés. Le protocole TAD-MAC apporte dans tous les cas un gain important. 19
16 Energie consommée par bit transmis avec succès pour différents niveaux de bruit Np. 20
17 (a): Variations temporelles du canal (b): energie dépensée correspondante. 22

18 Comparison of energy performance between an adaptive transmit power optimization versus the fixed transmit power at maximum level of *cc-2420* i.e. (0 dBm). 23

1.1 Sensor networks consisting of small nodes with an ability to sense the physical environment and communicate the information to the local (more powerful) base station. 26
1.2 An application context of GEODES Project. 27
1.3 Fire fighter scenario within a building (GEODES Project). 28

2.1 Reduced OSI model of wireless sensor networks for transmission (up link) and reception (down link). 35
2.2 General mechanism of preamble sampling protocols. The communication can either be initiated by receive or transmit node. If the communication is initiated by the receiver then the receive node will send the preamble to the transmit node and the transmit node will respond with the data packet, whereas if it is initiated by the transmitter then the preamble will be sent by the transmit node followed by the data. 36
2.3 BMAC protocol. 37
2.4 X-MAC protocol. 38
2.5 WiseMAC protocol. 39
2.6 Generic framework: the parametric-based cross layer interaction between the network, link/medium access control and the physical layers for the given application. Performance metrics such as $SINR$, BER, $RSSI$, etc., can help to adjust different parameters such as FEC selection, *Wake up Intervals*, *Adaptive Modulation*, *Dynamic Precision* etc. 41
2.7 (a) MAC And Physical LAyer Power (MAPLAP) Model. (b) State-based wireless sensor networks. 43
2.8 Component-based power model for WSN. 44
2.9 Basic Block Diagram of Wireless Sensor Node. 46
2.10 WSNet/WSIM simulator integrated design flow extracted from [27]. . . . 51

3.1 RICER3 is a 3-way handshake receiver initiated cycled receiver scheme. The N-data slots are placed in order to avoid the collision which happens when two nodes send the data to the same destination node at the same time in response to the wake-up-beacon. $\mathbb{E}[w]$ is the expected waiting time, T_{wu} is the total wake-up time of the receiver, I_{wu} is the wake-up interval of each node, WUB is the wake-up-beacon signal and Ack is the acknowledgment signal. 58
3.2 Hardware block diagram of PowWow. The key components in terms of power consumption are processing units (T.I MSP430, Actel IGLOO FPGA), DC/DC converter (used for voltage scaling), and the radio chip (T.I. cc2420). 65

3.3 The Control Packet (wake-up-beacon and acknowledgment) consists of geographical network address (x,y) of the receive node which transmits the control packets, cyclic redundancy check (CRC) and the last byte contains the type of packet (3-bits) and the mode of communication (5-bits). The data packet consists of the network address of source and destination as well as the hop-addresses of the next and last hop-node, along with few bytes for acknowledgment number, temperature, battery, and $N.T$ is frame buffer indicator. 66
3.4 Optimized 3-way handshake receiver initiated cycled receiver scheme. The N-Data slots are not necessary to avoid the collision, instead the receiver only requires the first byte of the data packet. Therefore, the idle channel monitoring is irrespective of data slots but adjustable according to the application. 67
3.5 The platform setup for measurements is constituted of three WSN Pow-Wow motes connected to the DC Power Analyzer. The external power supply of 3.3 *volts* is used. Four modules of the $N6705A$ are connected with the $cc2420$ and $MSP430$, to measure the current consumption of each component of the connected nodes as shown in Fig. 3.10. 68
3.6 Total current consumption: two nodes want to transmit the data to the same destination node. In this particular case, after 5 *blocks*, the transmitter wakes up and sends its WUB, and a rendez-vous (detailed in Fig. 3.8) occurs between the two nodes on the single channel. The total current consumption including the $cc2420$ and $MSP430$ is $22.5mA$. 69
3.7 Same as Fig. 3.6, only three nodes are presented separately for clarity. . . 70
3.8 Details of a rendez-vous: three phases i.e. WUB, DT and ACK are clearly shown, and all the events are described during the rendez-vous. The Tx *node* 1 senses the channel first and hence occupies the channel before the Tx *node* 2. Therefore the collision is being avoided. 71
3.9 Same as Fig. 3.8, only three nodes are presented separately for clarity. . . 72
3.10 Details of software and hardware consumption: the software component i.e. $MSP430$ consumes 0.0 μA, 0.01 μA, 2.5 mA, and 3.7 mA, in sleep, low power mode (LPM), transmitting, and receiving modes respectively. Whereas, the hardware component i.e. $cc2420$ consumes 0.0 μA, 0.4 mA, 17.6 mA, and 19.6 mA, in sleep, low power mode (LPM), transmitting, and receiving modes respectively. 73
3.11 Unsuccessful communication because of data collision, which is one of the bad executing scenario as mentioned in the energy model. This results in extra power consumption because of the waiting of ACK for the transmit nodes and data packet for the receive node. After about 4 msec. of wait in monitoring state, all nodes go to sleep mode. 74
3.12 Communication phase between transmit *nodes* and receive *nodes*: The energy model is based on execution traces of different scenarios which occur during the communication. These scenarios include: CBT (calculation before transmission), TWU (transmitter wake-up), WUB (wake-up-beacon), WUC (wake-up-collision), DT (data transmission), DR (data received), DC (data collision), DRE (data received with errors) and ACK (acknowledgment of data). 75

3.13 Hybrid Energy Model. The model utilizes the real-time measurements with an identification of typical scenarios that can occur in asynchronous WSN communication. By the help of the analytical model which evaluates the expressions for number of re-transmissions, number of wake-up-collisions and number of data collisions, the total global cost and the total energy consumption is evaluated. 78
3.14 Two nodes A and B are intending to transmit the data to the node C at a specific rate, and the energy consumption of node C is monitored. . . . 80
3.15 Test 1, Test 2 and Test 3 correspond to fixed parameters settings of I_{wu}, T_{wu}, T_{obs}, and TWU as $100ms$, $5ms$, $300s$, $2s$, whereas the data reception at node C varies from $1s$, $500ms$ and $100ms$. Y-Axis represents the energy consumed in mJ for an observation time of $300s$. 81
3.16 Total energy per successfully transmitted useful bit between nodes i and j as a function of the distance $D_{i,j}$ and of the transmission power level P_{Level}. 84

4.1 Traffic-aware dynamic MAC protocol is divided into two phases; the evolution phase before convergence (a) and the steady state phase after convergence (b). It is important to note that in the second phase the receive node (coordinator) has adapted its I_{wu} schedule as such that the idle listening is reduced. 92
4.2 Traffic-aware dynamic MAC protocol consists of two components (traffic estimation and adaptive wake-up interval algorithm). 93
4.3 Traffic status register bank: it contains N registers for N neighbor nodes. If the receive node receives data, the register (of the corresponding transmit node) is filled with $'1'$ and if it does not receive data, the TSR is filled with $'0'$. 94
4.4 Adaptive wake-up interval system. 97
4.5 System model used to characterize the adaptive wake-up interval algorithm. 100
4.6 The open loop form of the system consists of two parts. The first one is a non-adaptive system that compute ψ (for the purpose of analyzing controlled variables) but does not used it to update the wake-up interval whereas, the second part is real-time system which is the wireless sensor network. 101
4.7 Variations of mu variable against different (fixed) wake-up interval values. 102
4.8 Behavior of e variable against different (fixed) wake-up interval values calculated through correlation technique. First is the mean of the absolute value of correlation and second is only the mean of correlation. 103
4.9 Variations of e variable against different (fixed) wake-up interval values computed by counting number of ones in TSR. 104
4.10 Closed-loop form of the system considers the influence of feedback into the computation of the update of the wake-up interval in the adaptive system. It is to note that only μ and e are considered as controlled variables in the adaptive system. 105
4.11 Wake-up interval adaptation for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one, also showing the behavior of convergence under variable traffic. 106
4.12 Behavior of mu for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one. 107

4.13 Behavior of wake-up interval adaptation for number of different initial wake-up intervals. Few initial wake-up interval values result in oscillation and hence converge much later than others.The top figure is a zoomed version of the bottom one. 107
4.14 Variable *mu* behavior for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one. 108
4.15 Behavior of variable *e* for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one. 109
4.16 State machine being used to simulate TAD-MAC protocols in WSNet simulator. The scheduler moves to the next state based on the occurrence of an event, otherwise it keeps on waiting until the timer expires and then it proceeds backward. 111

5.1 Typical body area sensor networks consist of few nodes connected on the body and a coordinating node to control the communication between the nodes as well as with access points. 121
5.2 Hybrid network topology: invasive sensor nodes are connected with a coordinator through star network, whereas non-invasive nodes are configured through mesh topology. 122
5.3 Receive node adaptation with respect to transmit nodes based on the wake-up interval for a fixed data rate i.e. 1 s. It is to mention here that the reference clock's increment is 1 (ms), therefore all the time values are in ms. 125
5.4 Convergence of wake-up interval towards a steady state value for variable sensing rates. The traffic rate for node 1 and node 2 changes from 1 s to 1.5 s and 500 ms respectively. 125
5.5 Zoomed version of Fig. 5.4 to show convergence after changing the traffic rate. It can be seen that the receive node converges very efficiently as soon as the traffic changes. In this particular example, the traffic rate for node 1 and node 2 changes from 1 s to 1.5 s and 500 ms respectively. Node 1 converges to a steady wake-up interval value of 750 ms, whereas node 2 converges to 250 ms for the traffic pattern of [10101010...]. 126
5.6 Optimization of the TSR length and the corresponding convergence speed for different data transmission rates, starting from two packets per second to one packet in ten seconds. It can be seen that the convergence speed increases as the rate reduces, this is due to the fact that with reduced rate TSR is filled slowly and it results in slow convergence speed. 127
5.7 Lifetime prediction of AA alkaline battery is evaluated for various MAC protocols and radio chips. TAD-MAC along with *amis*52100 or *cc*1000 radio chips have major difference in comparison with other MAC protocols and *cc*2420. It is noteworthy to mention here that for low traffic rate *amis*52100 and *cc*1000 are preferable but for higher traffic rate *cc*2420 is more suitable. 132

6.1 Power profile characteristics of different radio transceivers operating at 2.4GHz and 900MHz. The curves show the current consumption against various power levels of the radio. 141
6.2 Energy per successful transmitted bit and corresponding bit error rate versus different transmit power levels under fixed noise (-90dBm). 143

6.3 Energy consumption under fixed noise power (−90dBm) against different transmit powers. 144
6.4 Different variants of varying noise power based Channel profiles. The limits of variations of the noise power are according to the maximum power that is available in most widely used radio transceivers for WSN. . 145
6.5 Energy consumption versus transmit power under varying but fixed noise power Np. 147
6.6 Execution flow of adaptive transmit power control technique. 148
6.7 Energy consumption versus number of different transmit power values. Optimal points on each curve is identified and can be seen by a dark black line. Whereas, the outlined curve (in red) represents the best posssible result for energy efficiency. 149
6.8 Best possible energy efficiency versus transmit power levels of *cc-2420* after applying the adaptive transmit power optimization under the channel variations of *model-c*. 150
6.9 (a): Channel variations 'model-c'. (b): Performance of the adaptive transmit power technique against the channel variations of *model-c* over time. 151
6.10 Best possible energy efficiency versus transmit power levels of *nRF-900* after applying the adaptive transmit power optimization algorithm under the channel variations of *model-d*. 152
6.11 (a): Channel variations 'model-d'. (b): Performance of adaptive transmit power algorithm against the channel variations of *model-d* over time. . . . 152
6.12 Comparison of energy performance between an adaptive transmit power optimization versus the fixed transmit power at maximum level of *cc-2420* i.e. (0 dBm). 154
6.13 Comparison between an adaptive transmit power optimization versus the fixed transmit power at maximum level of *nRF-900* i.e. (10dBm). 155

List of Tables

2 Évaluation de la précision de notre modèle. 14

2.1 Wireless Sensor Platforms. 48

3.1 System parameters of the physical platform and the timing values of the MAC Protocol. T_{react} is the reaction time for switch the states, T_{sync} is the time to synchronize, $T_{DF}(Rx)$ is the time to receive the data frame, $T_{DF}(Tx)$ is the time to transmit the data frame. 69

3.2 Software and hardware energy consumptions (Joules) of the executing scenarios which are identified during communication between sensor nodes. The energy consumption, in different states (tx, rx, $link$, and $ntwk$), and of the associated components *timer* and *controller* of the software component (microprocessor $MSP430$) are presented. For the case of hardware component (radio transceiver $cc2420$), energy consumption in transmit ($cc.Tx$) and receive ($cc.Rx$) states are presented. 79

3.3 Performance evaluation of validated results. 81

3.4 The cost of collision energy (C.Energy) results in an extra power consumption of the individual nodes and consequently of the complete network. The traffic is generated for different I_{wu} and accordingly the probability of collisions are calculated to compute the collision energy. 83

4.1 The number of beacons and acknowledgments transmitted by the receive node and number of data packets received by the receive node before reaching the convergence for sensing rate of 1s. Three cases (average, best and worst) are presented based on the performance of three (average, best and worst) initial wake-up interval values. 113

4.2 The number of beacons and acknowledgments transmitted by the receive node and number of data packets received by the receive node before reaching the convergence for sensing rate of 500ms. Three cases (average, best and worst) are presented based on the performance of three (average, best and worst) initial wake-up interval values. 114

4.3 Performance metrics for sensing rates of 1s before reaching a steady state. 114

4.4 Performance metrics for sensing rates of 500ms before reaching a steady state. 115

4.5 Total energy consumed by a receive node before reaching towards a steady state for sensing rate of $1s$ and $500ms$. Tab. 4.1 corresponds to sensing rate of $1s$ and Tab. 4.2 corresponds to $500ms$ and are used to calculate the energy consumption for average, best and worst cases. 117

4.6 Total energy consumed by a transmit node before reaching towards a steady state for transmission rate of 1 *packet/s* and 2 *packet/s*. Tab. 4.3 and Tab. 4.1 is used for evaluating the energy consumption of idle listening and data transmission for 1 *packet/s*. Tab. 4.4 and Tab. 4.2 are used for 2 *packet/s*. 117

5.1 The length of the traffic status register is optimized according to the fastest convergence speed towards a steady state value of the wake-up interval for variable traffic. 127
5.2 Parameter specifications of three widely used radio chips for WSN/WBASN. 129
5.3 Energy consumption (for one complete communication) evaluated for different radio chips and for different MAC protocols. The time spent by protocols in *Tx*, *Rx* and *Idle* states are evaluated through *WSNet* network simulator. Further by using the characteristics of different radio chips (shown in Table 5.2) energy consumption is estimated. 130
5.4 Physiological data being monitored and their respective data rates. 133
5.5 Energy consumption of the TAD-MAC protocol under variable traffic of WBASN for three different radio chips. The energy consumption for one complete communication is less than 0.3 *mJ* for *cc*2420 and is less than 0.2 *mJ* for *amis*52100 and *cc*1000. The time spent by the transmit node in idle listening is only 3.2 *ms*. 133

6.1 Performance evaluation of adaptive versus fixed transmit power under channel variations of *channel-c* for *cc-2420* radio transceiver. 155
6.2 Comparison after 1 *hr*. of transmission with packet rate of 1 packet/s under channel variations of *channel-c* for *cc-2420* radio transceiver. . . . 156
6.3 Performance evaluation of adaptive versus fixed transmit power under channel variations of *channel-d* for *nRF-900* radio transceiver. 156
6.4 Comparison after 1 *hr*. of transmission with packet rate of 1 packet/s under channel variations of *channel-d* for *nRF-900* radio transceiver. . . . 156
6.5 Look-up-table for an optimized implementation of adaptive transmit power technique. 157

Abbreviations

ASIC	Application Specific Integrated Circuit
ARQ	Automatic Repeat Request
ACK	Acknowledgment
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BMAC	Berkeley MAC
BNC	Body Network Coordinator
CBT	Calculation Before Transmission
CCA	Clear Channel Assessment
CRC	Cyclic Redundancy Check
DASN	Distributed Asynchronous Sensor Networks
DC	Data Collision
DLL	Data Link Layer
DVFS	Dynamic Voltage and Frequency Scaling
ECG	ElectroCardioGram
EMG	ElectroMyoGrapm
EEG	ElectroEncephaloGrapm
ECC	Error Correcting Codes
ETX	Expected Transmission Count
FEC	Forward Error Correction
FIFO	First In First Out
GEODES	Global Energy Optimization for Distributed Embedded Systems
HARQ	Hybrid Automatic Repeat Request
ICU	Intensive Care Unit
ISM	Industrial Scientific and Medical
LQI	Link Quality Indicator
LQE	Link Quality Estimation
LUT	Look-Up-Table
MAC	Medium Access Control
MAPLAP	MAC And Physical LAyer Power

$MEMS$	Micro-Electro-Mechanical-Systems
$MICS$	Medical Implant Communication Service
N	Number of Data Slots
OR	Occupancy Rate
PER	Packet Error Rate
PHY	Physical Layer
PL	Packet Length
$PW-MAC$	Predictive Wake-up MAC
$RICER$	Receiver Initiated CyclEd Receivers
$RI-MAC$	Receiver Initiated MAC
$RSSI$	Received Signal Strength Indicator
RNP	Required Number of Packets Re-transmission
RT	Re-transmission Rate
SNR	Signal-to-Noise Ratio
$TAD-MAC$	Traffic-Aware Dynamic MAC
$TDMA$	Time Division Multiple Access
$TICER$	Transmitter Initiated CyclED Receiver
TSR	Traffic Status Register
$WBASN$	Wireless Body Area Sensor Networks
$WiseMAC$	Wireless Sensor MAC
$WLAN$	Wireless Local Area Networks
$WMEWMA$	Window Mean Exponential Weighted Moving Average
$WMTS$	Wireless Medical Telemetry Service
WUB	Wake-up Beacon
WUC	Wake-up Collision
WSN	Wireless Sensor Networks

T_{wu}	Wake-up time
I_{wu}	Wake-up interval
T_{obs}	Observation time
tol	Tolerance
pl	Payload
P_b	Bit error probability
n_r	Number of nodes within radio range
Pr_{PC}	Probability of *preamble collision*
Pr_{DC}	Probability of *data collision*
G_r	Gain of the receiver antenna
G_t	Gain of the transmitter antenna
P_{Level}	Transmit power level
E_{tot}	Total energy consumption
E_{algo}	Energy consumed by the microcontroller
E_{proc}	Energy consumed by the processing (analog and digital)
E_{amp}	Energy consumed by the power amplifier
C_B	Basic cost
C_{RT}	Te-transmission cost
C_{DRE}	Cost of data received with errors
$\mathbb{E}[w]$	Expected wait (Idle listening)
$E(t)$	Error computation
X_1	First half of the status register
X_2	Second half of the status register
$N_{0,i}$	Number of zeros in i_{th} half
$N_{1,i}$	Number of ones in i_{th} half
$Nc_{0,i}$	Consecutive pairs of zeros in i_{th} half
$Nc_{1,i}$	Consecutive pairs of ones in i_{th} half
L	Length of the traffic status register
α	Constant weighting factor
μ	Update factor
t_{ref}	Reference time (system/simulator)
TSR_{ref}	Reference traffic status register
h	Hamming distance
e	Correlation error
ψ	Output of controlled variables in open and closed-loop forms
$nRF-900$	Nordic Semiconductor Radio Transceiver
Eb	Energy per bit

Résumé étendu

0.1 Introduction

Les réseaux de capteurs représentent un terrain de recherche et d'expérimentation privilégié de l'équipe CAIRN, notamment pour les contraintes énergétiques inhérentes à ce type de réseau, qui obligent à concevoir des architectures et des algorithmes très efficaces. Dans le domaine des réseaux de capteurs (RdC), les contraintes énergétiques sont telles qu'il n'est plus possible d'avoir une approche d'optimisation monocouche, il faut plutôt s'orienter vers une optimisation multi-couches ou dite *cross-layer*. Sans aller jusqu'aux protocoles de routage, dont l'influence sur la consommation globale du réseau est indéniablement très importante, mais qui ne rentrent pas vraiment dans le cadre de nos travaux plutôt consacrés au matériel mais surtout aux couches protocolaires les plus basses, les travaux de cette thèse ont pour but d'élaborer une méthodologie d'optimisation énergétique et de proposer quelques techniques novatrices au niveau algorithmique.

L'optimisation énergétique est un problème complexe puisqu'il fait appel à de très nombreux paramètres souvent interdépendants. Ces derniers peuvent être séparés en deux classes distinctes, les paramètres matériels qui dépendent de l'architecture du nœud de capteur et les paramètres logiciels qui dépendent des algorithmes de traitement, de stockage et de transmission mis en œuvre dans ce nœud, ainsi que de la topologie du réseau de capteurs et des protocoles utilisés. Alors qu'il y a quelques années les architectures de nœud de capteurs étaient vraiment simples, avec un microcontrôleur central basse consommation utilisé pour le traitement, le stockage et la transmission de données issues de capteurs périphériques analogiques, ce n'est plus vraiment le cas aujourd'hui. La modernisation des processus de fabrication des puces électroniques, l'émergence de modes d'alimentation intelligents — gestion dynamique de la tension d'alimentation,

veille, coupure de l'alimentation —, le développement de co-processeurs reconfigurables bas-coût et basse consommation permettent de définir de nouveaux modules plus performants, moins gourmands en énergie et dont la taille est de plus en plus réduite. Encore faut-il réussir à utiliser ce potentiel le plus efficacement possible du point de vue énergétique pour parvenir à une autonomie complète. La variété algorithmique semble encore plus vaste, puisque, pour chaque couche protocolaire, la recherche a fourni ces dernières années une telle quantité de solutions qu'il est très difficile de choisir l'ensemble le plus adapté à une problématique industrielle donnée.

Un autre point important est que l'optimisation énergétique ne peut se faire de façon efficace que si l'on est capable d'évaluer très précisément la consommation énergétique lorsque les nœuds de capteurs et le réseau global utilisent un jeu de paramètres particulier. Cette consommation énergétique est généralement divisée en deux parties : l'énergie due aux traitements numériques, c'est-à-dire celle qui est utilisée par les différentes unités de traitement ou de stockage, et l'énergie utilisée pour la transmission sans-fil de l'information vers les nœuds à portée radio.

La prépondérance de l'énergie de transmission dans la consommation globale d'un réseau de capteurs est aujourd'hui unanimement reconnue, et c'est la raison principale pour laquelle il est conseillé de transmettre à de faibles distances et d'utiliser les protocoles multi-étapes (ou *multi-hop*). Il faut donc chercher à minimiser autant que possible l'activité radio des nœuds. En ce qui concerne les protocoles MAC, nos travaux consistent à mettre en évidence, pour un protocole donné, l'influence des paramètres le caractérisant, et de les adapter à un trafic variable. Enfin, en ce qui concerne la couche physique, étant donné le panel très important d'algorithmes disponibles, la question que l'on peut se poser est la suivante : quelle quantité de traitement de signal peut-on ajouter pour diminuer l'énergie de transmission et optimiser la consommation énergétique globale?

Pour illustrer quelques unes de ces innovations, nous utilisons dans cette thèse des résultats obtenus sur la plate-forme de réseaux de capteurs sans fil développée par l'équipe-projet INRIA CAIRN qui intègre la plupart des nouveautés pré-citées. Cette plate-forme,

baptisée PowWow[1] *(Power Optimized hardWare and software frameWOrk for Wireless motes)*, combine une conception matérielle modulaire et ouverte à du logiciel libre à très faible empreinte mémoire reposant sur un système à programmation événementielle.

0.2 Méthodologie d'optimisation

Pour concevoir un réseau de capteurs (RdC) sans fil efficace en énergie, il faut être capable d'explorer l'ensemble des paramètres qui le caractérisent [33]. La méthodologie générale qui peut être appliquée pour cette exploration est représentée sur la Figure 1. Nous nous plaçons dans le cadre d'une topologie de réseau déjà établie, et non pas dans la phase de déploiement du réseau (il semble toutefois envisageable d'utiliser un modèle d'énergie encore plus complexe que celui que nous présenterons dans la section suivante pour produire une répartition optimale des nœuds du point de vue énergétique). Cette topologie nous donne donc des indications sur le nombre de nœuds, sur les distances mises en jeu entre les nœuds, sur le ou les types de canal de propagation qui seront empruntés par les signaux. Suivant l'application visée vont alors venir s'ajouter des contraintes de qualité de services tels le taux d'erreur binaire, le rapport signal sur bruit ou encore le délai maximal d'acheminement des données. Un nœud de capteur dispose idéalement d'un certain nombre de possibilités algorithmiques au niveau des couches basses protocolaires regroupées dans une bibliothèque. Il peut ainsi choisir différents schémas de modulation, de codage correcteur d'erreurs, d'algorithmes de compression des données, voire des techniques de coopération. Les protocoles d'accès au médium et de routage doivent également être pris en compte car ils vont décider notamment du nombre de transmissions opérées par un même nœud, et ce taux de sollicitation influence sa durée de vie de façon significative.

Une fois qu'une sélection est opérée, elle est fournie au simulateur de consommation énergétique qui, pour une application donnée, pourra évaluer la dépense énergétique en fonction de plusieurs modèles de consommation (énergie de circuit, énergie de transmission, modèle énergétique de processeur, etc.) qui sont présentés dans la section suivante.

[1] L'ensemble des composants de la plate-forme PowWow est disponible sur le site du projet `http://powwow.gforge.inria.fr/`.

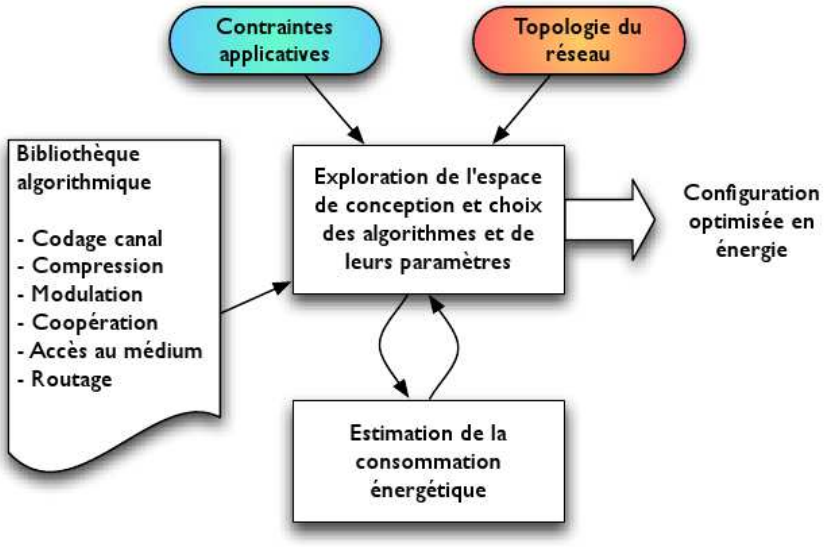


FIGURE 1: Méthodologie d'optimisation par exploration de paramètres

Ce processus est répété pour plusieurs configurations et le sélectionneur de configuration retiendra finalement la configuration optimale en énergie.

0.3 Modèle de consommation d'énergie

En règle générale, le modèle d'énergie de haut-niveau adopté est défini par la somme entre l'énergie de "calcul" consommée par les divers traitements et l'énergie dédiée à la communication entre les capteurs. Cette première décomposition n'est pas extrêmement claire, étant donné que la frontière entre les traitements et les communications n'est pas facile à mettre en évidence. La part de communication prend-elle en compte l'intégralité de la gestion de la pile de protocole réseau ? Cette distinction est possible, mais dans ce cas de nombreux traitements numériques ainsi que l'exécution de protocoles complexes tels que le routage ou encore la fiabilisation des données par la couche liaison sont alors inclus dans le terme d'énergie de communication, alors que leur estimation a plus de similarité avec l'évaluation de la consommation des traitements [25].

Une deuxième possibilité de décomposition consiste à distinguer les traitements numériques des traitements analogiques. L'énergie totale E_{tot} peut alors s'exprimer comme

$$E_{\text{tot}} = E_{\text{algo}} + E_{\text{analog}} + E_{\text{amp}}$$

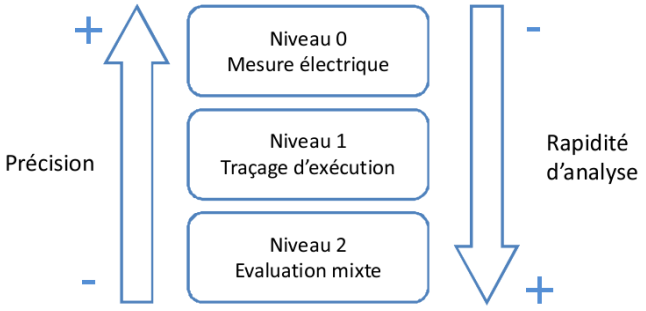


FIGURE 2: Méthodes d'évaluation de la consommation d'énergie

avec E_{algo} l'énergie due aux calculs numériques, E_{amp} l'énergie consommée par l'amplificateur d'émission analogique et enfin E_{analog} l'énergie due aux traitements analogiques de mise en forme des signaux de communication et toute autre cause analogique de consommation. En considérant la plate-forme choisie, le terme E_{algo} représente la consommation du processeur et de ses mémoires RAM et flash, ainsi qu'une part de traitement numérique qui serait incluse dans la partie radio. Le terme E_{analog} correspond quant à lui à l'énergie dépensée par le capteur et le convertisseur analogique-numérique et l'énergie de la partie radio analogique à l'exception de l'amplificateur. Le terme E_{amp} est associé à l'amplificateur présent dans la partie radio. Cet élément est à part parce qu'il constitue à lui tout seul une part importante de la consommation et est fortement variable dans un contexte de réseau de capteurs où l'énergie est fortement contrainte.

Pour évaluer la consommation énergétique, une approche consiste à considérer trois niveaux d'estimation (Figure 2). Le niveau 0 consiste à mesurer le produit courant tension aux bornes de l'alimentation pendant le fonctionnement du système. Cette technique d'évaluation aussi précise et simple soit-elle, est difficile à appliquer en pratique et souffre d'inconvénients majeurs dans la plupart des cas. Pour réaliser cette mesure, il suffit d'un oscilloscope numérique à mémoire capable de fournir les données recueillies sous forme de fichiers de données à exploiter par un logiciel de traitement. Si le protocole expérimental est relativement simple, il faut cependant pouvoir disposer du système matériel réel et complet entièrement défini et opérationnel, ce qui va à l'encontre d'un besoin de prototypage et d'anticipation nécessaire pour obtenir des systèmes atteignant les performances d'un cahier des charges donné.

L'évaluation de niveau 1 consiste à effectuer l'analyse à partir du traçage d'exécution de code. Il s'agit:

1. de classifier les comportements des capteurs sous forme de cycles d'exécution,
2. de mesurer la consommation énergétique pour chacun des cycles d'exécution et
3. de compter ces cycles d'exécution pendant le déroulement d'un programme sur la plate-forme réelle.

L'évaluation de niveau 2 se base sur l'analyse du niveau 1, en cherchant cette fois-ci à évaluer de façon analytique et statistique les comportements des capteurs:

1. la classification des cycles d'exécution est la même que pour le traçage d'exécution de code,
2. l'évaluation de consommation énergétique pour chacun des cycles d'exécution est en partie analytique,
3. le nombre des cycles d'exécution est évalué de façon analytique et statistique.

L'évaluation intermédiaire représente sans doute le bon compromis entre précision et complexité et c'est la voie abordée notamment dans [14]. Disposer d'un modèle énergétique de grande précision est d'une importance cruciale pour la conception et le déploiement d'un réseau de capteurs. Il permet à la fois d'exploiter les techniques innovantes au niveau architectural et de choisir les valeurs optimales du point de vue énergétique des nombreux paramètres des différentes couches protocolaires. Si la complexité reste raisonnable, le modèle peut même être utilisé en temps réel pour adapter dynamiquement certains paramètres, comme nous le verrons dans les sections suivantes.

0.3.1 Mesures expérimentales

0.3.1.1 Plate-forme PowWow

PowWow *(Power Optimized hardWare and software frameWOrk for Wireless motes)* [23, 63, 106] est une plateforme matérielle et logicielle conçue pour mettre en œuvre des réseaux de capteurs fortement contraints en énergie. Elle est basée sur des protocoles MAC optimisés (cf. section suivante), un routage géographique simple et un code logiciel très compact (pile protocolaire d'environ 6 ko et application typique entre 4 et 5 ko). Des réductions en consommation avec des facteurs de plus de 15 ont été relevées par

rapport à une couche 802.15.5/Zigbee libre de chez Texas Instruments (TIMAC) pour des collectes régulières de température dans un réseau d'une dizaine de nœuds.

D'un point de vue du matériel, la plate-forme PowWow possède également quelques originalités. Si elle est bien sûr basée sur des composants standards (microcontrôleur, *transceiver* radio) comme nombre de plateformes, sa conception modulaire et son faible facteur de forme lui permetent d'empiler plusieurs couches de cartes selon les besoins applicatifs. L'architecture globale et les interactions entre les unités de traitement et de calcul sont illustrées par la Figure 3, tandis que la Figure 4 montre une photo de la plate-forme PowWow avec trois cartes (calcul, radio et collectage d'énergie). Quatre modules sont actuellement disponibles.

- La carte mère intègre tous les composants nécessaires à l'exécution de la pile protocolaire PowWow, excepté le composant radio. En particulier, celle ci intègre un microcontrôleur faible consommation MSP430 de Texas Instruments [119] qui est le composant central du système et fait office de contrôleur principal et de composant de calcul. Il s'agit de la version MSP430F1612 (55ko de mémoire flash et 5ko de RAM). Il consomme 330μA à 1 MHz et 2.2 V en mode actif et 1.1μA en mode veille. La carte intègre aussi le gestionnaire d'horloge (par défaut à 8 MHz) et d'alimentation (par défaut à 3V) permettant une gestion dynamique de la fréquence et de la tension d'alimentation. Des interfaces JTAG, RS232 et I2C sont disponibles sur cette carte pour interfacer avec un ordinateur ou des capteurs.

- La carte radio intègre un composant radio CC2420 [55] compatible Zigbee/802.15.4 et une connexion pour une antenne. Son association à la carte mère crée le système minimum permettant à PowWow de fonctionner.

- Entre ces deux cartes, il est possible de connecter une carte FPGA qui permet une accélération matérielle pour certains traitements de PowWow. Actuellement la version de la plate-forme intègre un FPGA faible consommation Actel Igloo [7]. Le composant, un AGL125 (125000 portes equivalentes, 32 kbits de mémoire RAM, 1 kbits de mémoire flash), consomme 2.2μW, 16μW et de 1 à 30mW respectivement dans les modes veille, *freeze* et actifs selon le taux d'utilisation.

- Enfin, une dernière carte fille est dédiée aux techniques de récupération d'énergie. Basée sur un composant de gestion de l'énergie LTC3108 de chez Linear Technology,

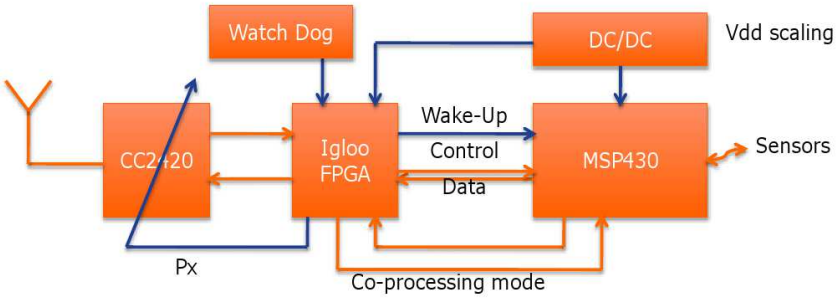


FIGURE 3: Architecture matérielle de la plate-forme PowWow



FIGURE 4: Photo de la plate-forme PowWow avec trois cartes: calcul, récupération d'énergie et radio

elle permet d'être configurée avec plusieurs types de stockage d'énergie (batteries, micro-batteries, super-capacités) et plusieurs types de sources d'énergie: un petit panneau solaire pour récupérer l'énergie photovoltaique, un capteur piézoélectrique pour l'énergie mécanique et un capteur d'énergie thermique de type Peltier.

0.3.1.2 Couche MAC asynchrone

La couche d'accès au Medium (MAC pour Medium Access Control) a pour rôle de permettre aux nœuds d'un réseau de partager de la manière efficace le ou les canaux sans fil disponibles. Son principal objectif dans le contexte des réseaux de capteurs est de minimiser la consommation d'énergie en éteignant les modules radios aussi souvent que possible. Les principales sources de consommation au niveau MAC sont l'écoute passive inutile (ou idle listening), les en-têtes parfois trop longues (ou overheads), la réception de

paquets destinés à un autre nœud (ou overhearing) et bien sûr les collisions. Ces dernières peuvent avoir lieu lors de l'émission de n'importe quel paquet, que ce soit une trame de réveil (WUB pour Wake-up Beacon) ou l'envoi de données (DT pour Data Transmission). Pour atteindre la meilleure efficacité, ces artéfacts doivent être minimisés, mais pas tous simultanément. Si l'on cherche à diminuer l'écoute inutile et les collisions, il va falloir augmenter les processus de synchronisation et les overheads. Inversement diminuer les overheads et la synchronisation résultera en des collisions plus nombreuses.

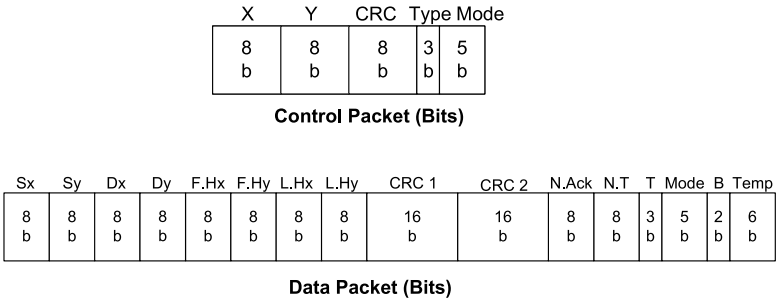


FIGURE 5: Les paquets de contrôle (WUB et ACK) comprennent l'adresse géographique du nœud qui les émettent, un code de redondance cyclique (CRC) et le dernier octet contient le type de paquet (3 bits) et le mode de communciation (5 bits). Le paquet de données contient les adresses de la source, de la destination, du dernier et du prochain nœud en cas de multi-étapes, ainsi que quelques octets pour le numéro d'ACK, les informations des capteurs, telle la température, le niveau de batterie et le numéro de trame.

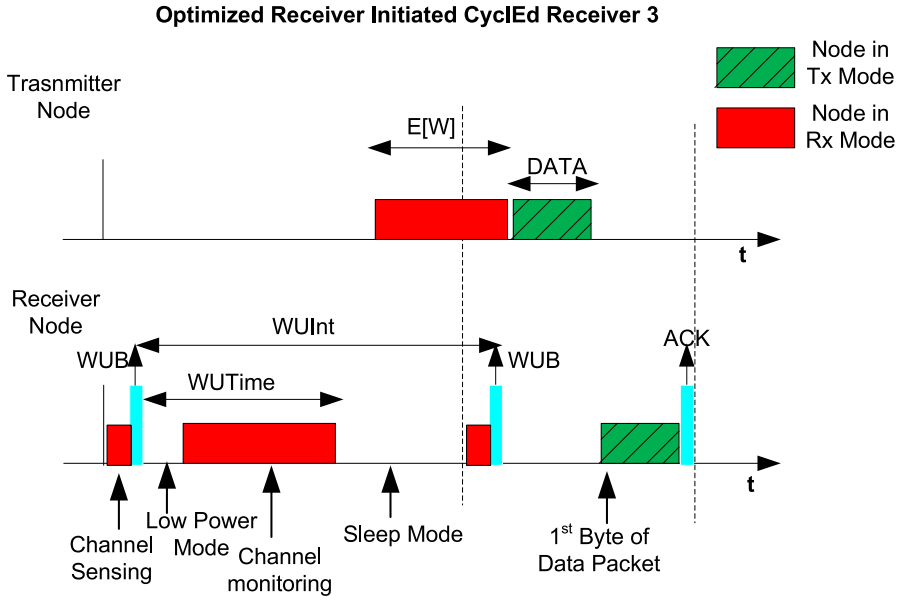


FIGURE 6: Protocole RICER optimisé ajustable à l'application et diminution de l'écoute inutile.

La famille à échantillonnage de préambule est une option intéressante pour les réseaux de capteurs et son mécanisme peut être initié soit par l'émetteur soit par le récepteur [19].

Dans le premier cas le préambule est envoyé par l'émetteur, directement suivi par les données. Le récepteur se réveille à intervalle régulier pour sonder le canal, et se rendort immédiatement s'il ne détecte aucun préambule. Si la communication est initiée par le récepteur, c'est ce dernier qui va envoyer régulièrement un préambule ou trame de réveil (WUB), restera en écoute pendant un instant et l'émetteur répondra en envoyant ses données. Les performances énergétiques de cette dernière stratégie nous ont séduites [82], et c'est ce processus asynchrone que nous avons choisi pour notre plate-forme PowWow.

Pour le rendre plus efficace, nous avons optimisé le protocole RICER3 en diminuant la taille des paquets et les overheads, comme le montre les Figures 3.3 et 3.4, qui représentent respectivement les compositions des paquets et le protocole lui-même. Par rapport à la version d'origine de RICER, il faut noter l'absence de plusieurs slots temporels après l'intervalle de réveil, mais ceci n'augmente pas le taux de collision.

0.3.2 Expérimentations

Pour pouvoir affiner le modèle d'énergie, nous avons besoin de connaître la consommation des nœuds dans les différentes états qui composent notre protocole. L'analyseur de puissance *Agilent* $N6705A$ équipé de quatre canaux pouvant chacun mesurer le courant consommé par un appareil nous a servi à monitorer plusieurs capteurs durant un laps de temps assez important pour pouvoir observer les différents états, y compris des collisions. L'impact d'une collision sur la consommation d'un nœud est ainsi représenté sur la Figure 3.11.

Dans ce cas précis, deux nœuds cherchent à émettre en même temps à la même destination, écoutent l'activité sur le canal et l'estiment libre. Ils envoient donc leur donnée simultanément, ce qui crée une collision (DC), et se retrouvent à attendre un accusé de réception qui ne viendra jamais. A cause de cette collision, le récepteur est incapable de détecter une information valide et reste $4ms$ en écoute, soit trois fois plus que la normale. Au final, nous avons estimé que le gaspillage d'énergie dû à cette collision était de $0.37.10^{-3}J$ à chacun des émetteurs et de $0.3888.10^{-3}J$ au récepteur.

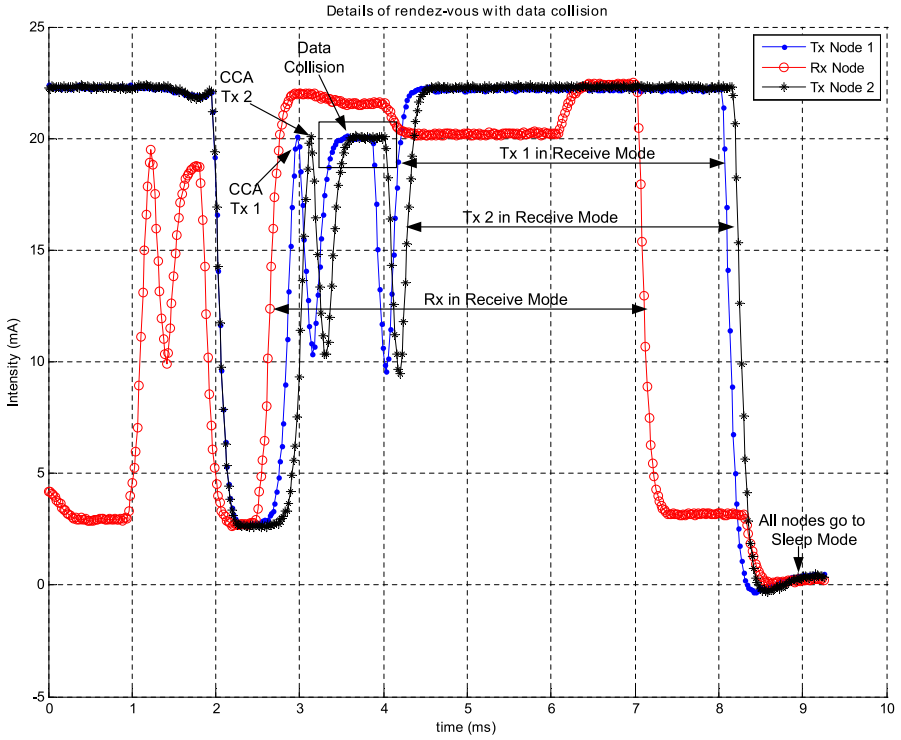


FIGURE 7: Communication ratée pour cause de collision de données entre les deux émetteurs.

0.3.3 Modèle hybride

Comme annoncé au début de cette section, les mesures expérimentales vont nous servir à alimenter un modèle hybride, dans lequel nous utilisons également des expressions analytiques en ce qui concerne les taux de transmissions réussies et de collisions.

Les différents états dans lesquels peuvent se trouver les nœuds au cours d'un échange de données sont représentés sur la Figure 3.12, et se définissent comme suit:

- Calcul avant émission (CBT)
- Réveil de l'émetteur (TWU)
- Envoi de la trame de réveil (WUB)
- Collisions de trames de réveil (WUC)
- Emission des données (DT)
- Réception des données (DR)
- Envoi d'un accusé de réception (ACK)

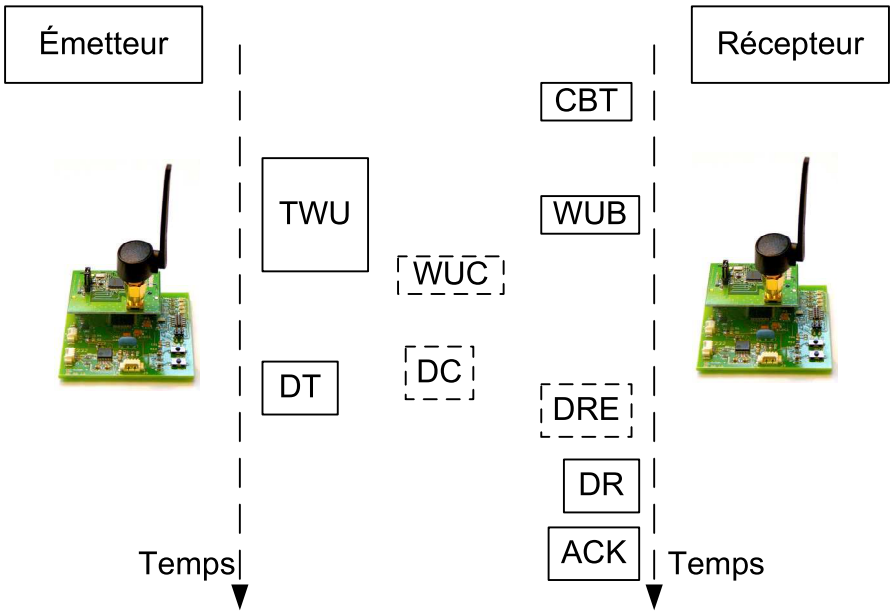


FIGURE 8: Phases de communication entre émetteurs et récepteurs. Le modèle d'énergie est basé sur les traces d'exécution des différents scénarios possibles durant la communication.

- Réception de données avec erreur (DRE)
- Collision de données (DC)

Les restangles en ligne continue sont les états classiques de toute transmission, tandis que les restangles en pointillés représentent des évènements imprévus qui peuvent arriver, notamment dans le cas de rendez-vous asynchrones.

Le principe de notre modèle d'énergie hybride [14] est présenté sur la Figure 3.13. En entrée, nous fournissons au modèle les différents paramètres temporels, tels l'intervalle de réveil $WUInt$, la durée d'un réveil $WUTime$, Le temps d'observation T_{obs}, le nombre de paquets de données λ et le nombre de nœuds dans le voisinage radio n_r. Le modèle utilise les mesures expérimentales sur la plate-forme PowWow qui ont permis de capturer la consommation d'énergie pour chacun des états décrits précédemment. Ensuite, avec l'aide d'expressions analytiques pour évaluer le nombre de retransmission N_{RT} et le nombre de collisions de trames de réveil N_{WUC} et de données N_{DC}, tous ces paramètres nous permettent de calculer le coût global en énergie.

Pour valider notre modèle, nous choisissons différents scénarios pour lesquels nous sommes capables de mesurer précisément la consommation d'énergie, et nous appliquons notre

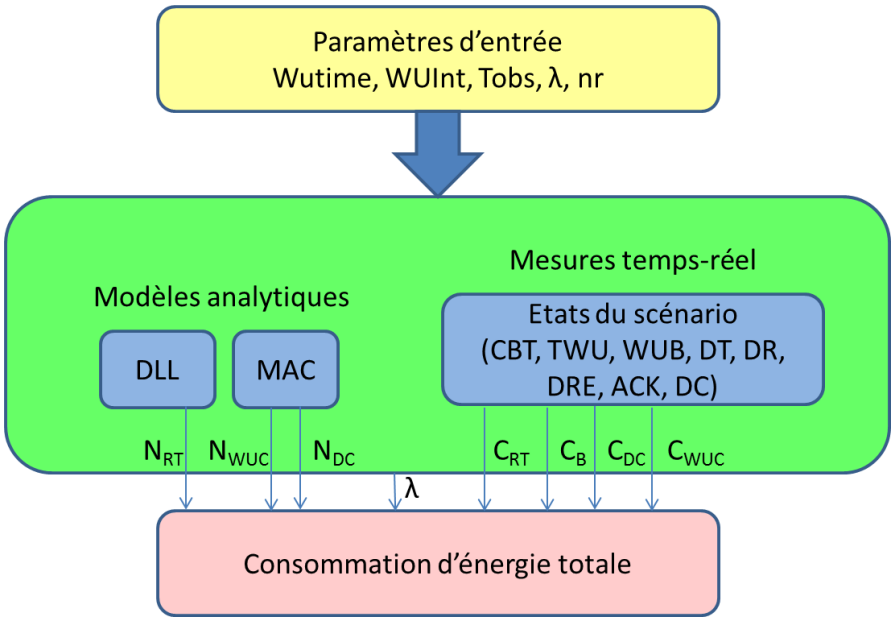


FIGURE 9: Modèle d'énergie hybride à partir de résultats expérimentaux et d'expressions analytiques.

modèle sur la même application [14]. Nous considérons ici deux nœuds qui veulent transmettre leur information respective à une même destination et monitorons cette situation pendant plusieurs minutes (nous sommes limités par la mémoire de l'analyseur de puissance, ce qui empêche des expérimentations de plusieurs heures). Pour éviter des cas particuliers les expérimentations sont répétées cinq fois dans la même configuration (A à E)

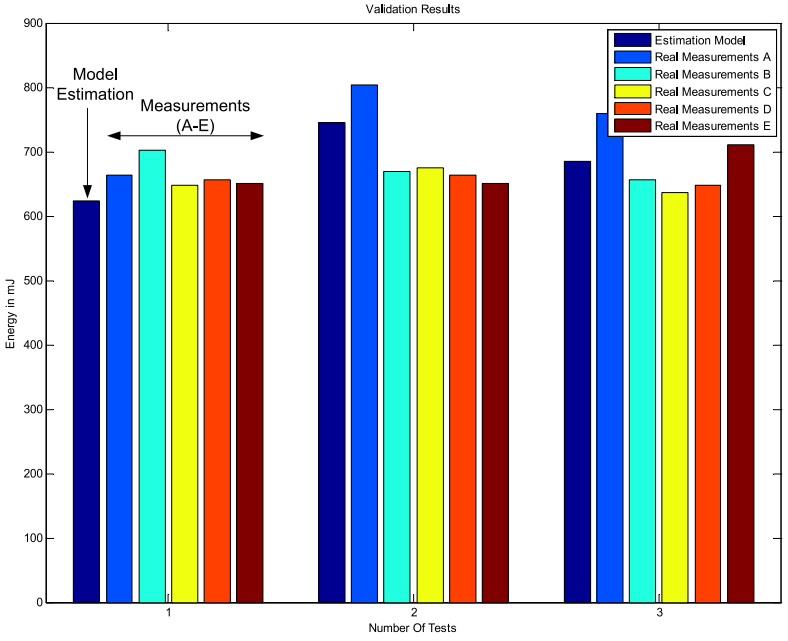


FIGURE 10: Comparaison de notre modèle d'énergie avec différentes sessions de mesures pour 3 tests de 300s avec des valeurs différentes pour TWU et $WUInt$.

Les résultats de ces tests, résumés dans la Table 3.3, démontrent clairement la précision de notre modèle, qui reste entre 1% et 8% d'erreur.

TABLE 2: Évaluation de la précision de notre modèle.

	Test 1	Test 2	Test 3
Estimation du modèle (mJ)	624.2	745.2	684.6
Moyenne des mesures (mJ)	664.2	692.2	682.2
Écart-type	19.59	56.36	46.18
Erreur Relative	6.0%	7.5%	0.35%

Le précision de ce modèle d'énergie nous permet d'optimiser le déploiement d'un réseau de capteurs en adaptant la topologie de notre réseau aux contraintes de l'environnement. Dans le cas d'un réseau déjà existant, nous pouvons régler certains paramètres physiques ou protocolaires pour s'adapter soit aux contraintes de l'environnement, soit à celles imposées par l'application. Ainsi, il est possible d'adapter la puissance d'émission en fonction de la distance de transmission ou plus efficacement en fonction du niveau de signal reçu (RSSI pour Receive Signal Strength Information) ou de la qualité de lien radio (LQI pour Link Quality Information), ou d'une estimation du RSB ou du TEB [51], comme nous le verrons dans la dernière section.

0.4 Adaptation dynamique de l'intervalle de réveil des capteurs

Dans de nombreux types de réseaux, il arrive que le débit soit très variable, et qu'une configuration donnée, bien qu'optimale pour un certain jeu de paramètres, devienne totalement inefficace à cause des variations de l'environnement. C'est le cas par exemple de certains réseaux de surveillance qui doivent faire transiter des images en cas d'alerte ou d'intrusion alors qu'un débit très faible était auparavant suffisant pour la maintenance du réseau. Nous nous plaçons dans cette section dans un autre contexte à débit variable, celui d'un réseau de capteurs corporels (WBASN pour Wireles Body Area Sensor Network) monitorant des données physiologiques. Ces données, provenant de capteurs intra ou extra-corporels, sont de natures très différentes : de simples capteurs de température ou de pression ne nécessitent par exemple pas autant de débit qu'un électrocardiogramme (ECG) ou un électroencéphalogramme (EEG) et le débit de certains de ces appareils varie

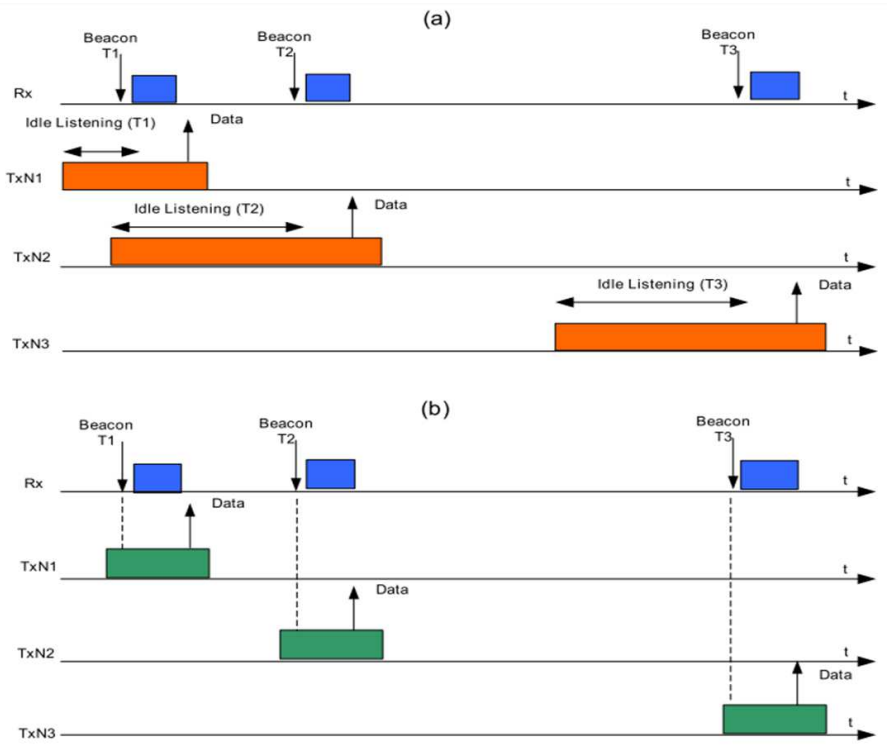


FIGURE 11: Adaptation de l'intervalle de réveil aux messages des différents capteurs

lui-même très fortement en cas d'alerte. Nous avons donc cherché à concevoir un protocole efficace en énergie, capable de s'adapter automatiquement à ces variations de débit, que nous avons baptisé TAD-MAC (Traffic Aware Dynamic MAC).

Le principe de ce protocole, présenté sur la Figure 11 consiste à ne réveiller le nœud central, qui est donc le récepteur (R) que lorsqu'une information lui étant destinée transite sur le réseau, évitant ainsi des écoutes inutiles (idle listening) très pénalisantes pour le bilan énergétique global. Dans un WBASN, plusieurs nœuds, ici symbolisés par TX1, TX2 et TX3, vont émettre leurs données à des temps différents et une périodicité différente, et le récepteur doit donc adapter ses instants de réveil à cette demande. La figure est divisée en deux parties, la première représente le fonctionnement du réseau avant convergence, tandis que la seconde présente un réseau ayant convergé.

Pour mettre en œuvre cette optimisation, les trames envoyées par le récepteur indiquent l'instant du prochain réveil. Lorsqu'un nœud se réveille mais que la trame de réveil affirme que c'est à une source concurrente de transmettre, il peut soit rester en attente

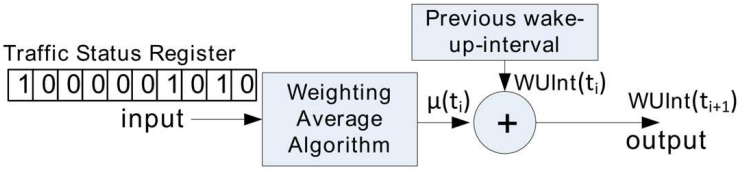


FIGURE 12: Algorithme de mise à jour de l'intervalle de réveil. TSR désigne le registre de statut du trafic, dans lequel '1' veut dire que le nœud a reçu un paquet qui lui était destiné, tandis que '0' révèle un réveil inutile.

si le prochain WB est imminent, soit retourner en veille et ne se réveiller qu'au moment opportun. Le processus de convergence s'opère grâce à un registre indiquant le trafic (TSR pour Traffic Status Register) que le récepteur tient à jour pour chacun de ses voisins. Ces registres à décalage insèrent un '1' lorsque'un message provenant de la bonne source a été correctement reçu, et un '0' lorsque le réveil s'est révélé infructueux. Le cas idéal serait évidemment celui où les TSR ne contiendraient que des '1', ce qui voudrait dire que tous les réveils sont utiles (inversement plus il y a de '0' et plus le système gaspille d'énergie), mais vouloir obtenir ce genre de configuration s'avère très dangereux puisque le récepteur ne serait pas capable de s'adapter à la moindre hausse de débit, et pourrait rater des informations capitales. Nous avons donc décidé de choisir comme cible un TSR dans lequel les '0' et les '1' s'intercalent régulièrement, de façon à pouvoir s'adapter très rapidement à des variations de débit tout en gaspillant peu d'énergie en écoutes inutiles [16]. Un algorithme utilise alors ces différents registres TSR pour adapter les réveils du nœud central à la demande des capteurs physiologiques, comme le montre la Figure 12.

Lorsqu'un nœud se réveille, il envoie sa trame de réveil puis calcule son nouvel intervalle de réveil à partir du TSR. Cette nouvelle valeur à l'instant $(t_i + 1)$ est reliée à $WUInt$ et à un facteur de mise à jour μ à l'instant (t_i) par

$$WUInt(t_{i+1}) = WUInt(t_i) + \mu(t_i) \cdot t_{ref}. \quad (1)$$

Pour n'avoir qu'une seule référence de temps pour $WUInt$ et μ, t_{ref} est multiplié par μ et est défini en fonction du l'horloge du simulateur. Le facteur de mise à jour $\mu(t_i)$

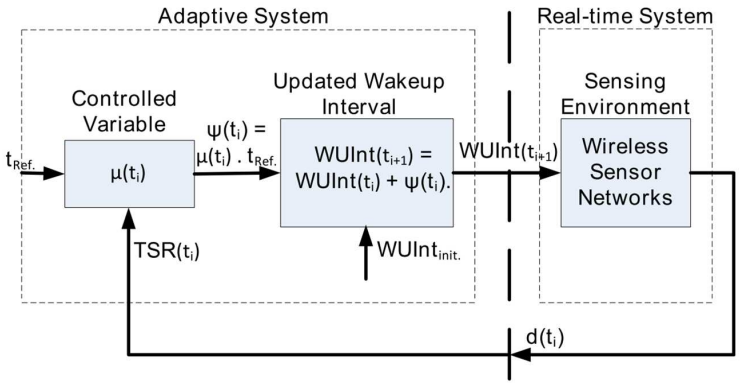


FIGURE 13: Vue d'ensemble du système adaptatif.

est calculé à partir des moitiés du TSR respectivement les plus significatives $(X_1(t_i))$ et moins significatives $(X_2(t_i))$ avec un facteur de pondération α :

$$\mu(t_i) = \alpha \cdot X_1(t_i) + (1-\alpha) \cdot X_2(t_i), \tag{2}$$

où $X_1(t_i)$ et $X_2(t_i)$ valent

$$X_1(t_i) = \frac{N_{0,1}}{L/2} \cdot Nc_{0,1} - \frac{N_{1,1}}{L/2} \cdot Nc_{1,1}, \tag{3}$$

$$X_2(t_i) = \frac{N_{0,2}}{L/2} \cdot Nc_{0,2} - \frac{N_{1,2}}{L/2} \cdot Nc_{1,2}. \tag{4}$$

Dans ces expressions, $N_{0,1}$, $N_{1,1}$, $Nc_{0,1}$ et $Nc_{1,1}$ sont respectivement le nombre de zéros, le nombre de uns, les nombres d'occurrences de zéros consécutifs et de uns consécutifs dans X_1, tandis que $N_{0,2}$, $N_{1,2}$, $Nc_{0,2}$ et $Nc_{1,2}$ sont respectivement le nombre de zéros, le nombre de uns, les nombres d'occurrences de zéros consécutifs et de uns consécutifs dans X_2. Le paramètre d'optimisation L représente quant à lui la longueur du TSR. Ce dernier peut en effet être ajusté en fonction du trafic, comme nous l'observons sur la Figure 14, pour des taux de transmissions variant de 2 paquets par seconde à 1 paquet toutes les 10 secondes. Pour de faibles trafics, la longueur du TSR optimale s'établit ainsi à 12 bits (longueur maximale que nous nous étions autorisés). La vitesse de convergence augmente elle aussi lorsque le trafic diminue, car le remplissage du TSR ralentit [15].

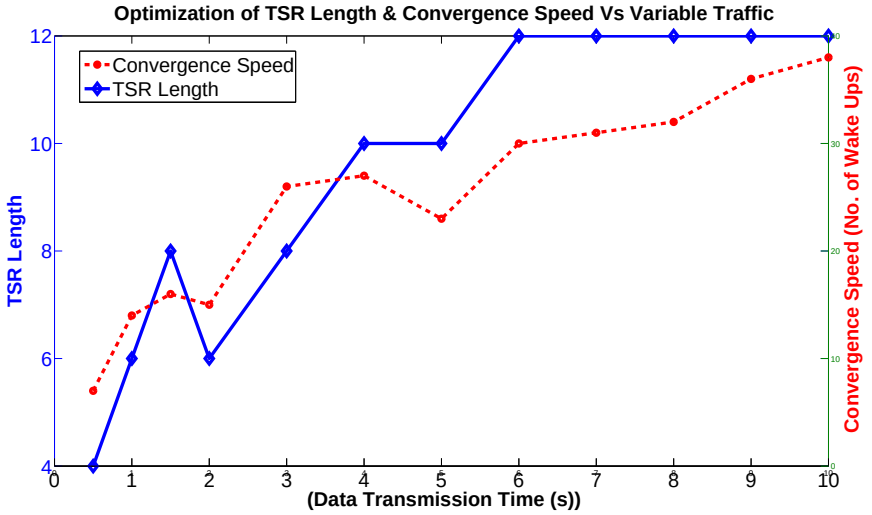


FIGURE 14: Optimisation de la longueur du TSR et vitesse de convergence associée en fonction du trafic (de 2 paquets par seconde à 1 paquet toutes les 10 secondes).

Pour pouvoir évaluer les performances de notre protocole, nous utilisons le simulateur de l'INRIA dédié aux réseaux de capteurs *WSNet* [27] [28] qui va nous permettre de connaître le temps passé par un élément du réseau dans différents états dont nous connaissons la consommation. La simulation consiste en un réseau de 10 nœuds dont l'un est un coordinateur au centre d'une topologie en étoile. Le trafic suit une distribution de Poisson et le canal est à évanouissements de Rayleigh. Le modèle applicatif impose un réveil toutes les 100 ms, un débit de 1 paquet/s et le scénario est joué durant 50 minutes [15].

Outre RICER, que nous avons déjà présenté, les protocoles MAC retenus pour cette évaluation sont BMAC, XMAC et WiseMAC. Dans BMAC [95], le nœud voulant transmettre place un long préambule avant les données pour que la destination reçoive celui-ci quand elle se réveille, ce qui n'est pas très efficace en énergie. XMAC [24] évite ce gaspillage en remplaçant ce long préambule par de multiples préambules plus courts. Dès que la source reçoit l'accusé de réception de la destination, elle peut transmettre ses données. Dans WiseMAC enfin, la source adapte ses instants de réveil à la destination, ce qui diminue encore les moments d'écoute inutile [59, 60] (la destination indique son prochain réveil dans son accusé de réception), mais le protocole n'est pas conçu pour des adaptations dynamiques.

La métrique illustrée ici par la Figure 15 est la durée de vie du réseau que l'on cherche bien sûr à maximiser (celle-ci est définie ici par l'instant où le premier nœud épuise sa batterie). Trois transceivers radio différents sont considérés, le $CC2420$ bien sûr, qui est celui utilisé par la plate-forme PowWow, mais aussi les $amis52100$ et $CC1000$. La tension d'alimentation de ces derniers est plus faible, ce qui explique leurs meilleurs résultats, mais ils travaillent dans la bande de fréquence ISM à 868 MHz (contrairement au CC2420 qui émet à 2.4 GHz) et ont un débit potentiel plus faible . Le gain en énergie apporté par TAD-MAC est vraiment remarquable, et varie de 3 à 18 en fonction du circuit utilisé et du protocole avec lequel on se compare. On peut noter que si le CC2420 est ici le transceiver le moins efficace, ce n'est plus le cas si le trafic devient plus intensif. Enfin, il faut bien comprendre que ces résultats sont pour un réseau statique, ce qui veut dire que l'avantage de TAD-MAC dans le cas d'un réseau à trafic variable serait bien plus important encore, mais la comparaison ne serait pas vraiment juste puisque les autres protocoles n'ont pas été conçus dans cette optique.

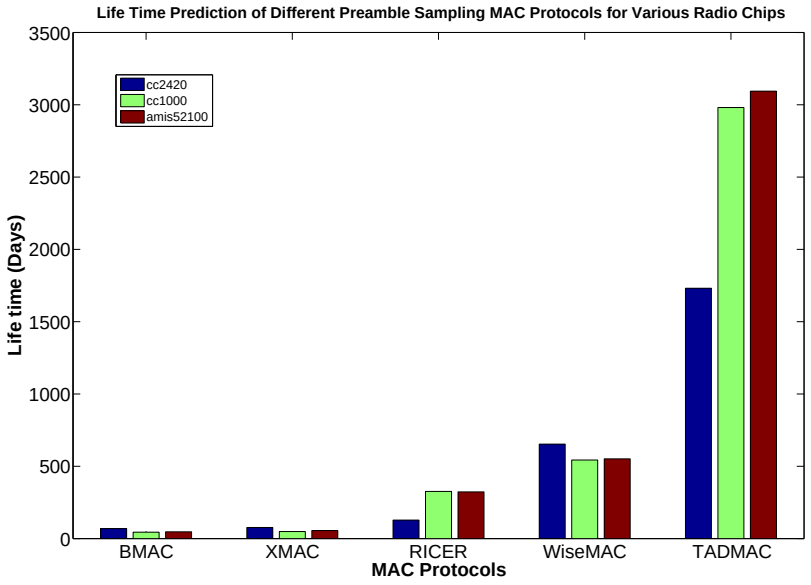


FIGURE 15: Durée de vie d'un nœud (batterie alkaline AA) pour différents protocoles MAC et transceivers radio. Les transceivers $amis52100$ et $cc1000$ sont préférables pour de faibles trafics tandis que le $cc2420$ convient à des trafics plus élevés. Le protocole TAD-MAC apporte dans tous les cas un gain important.

0.5 Adaptation dynamique de la puissance d'émission

En prenant l'hypothèse de conditions statiques à la fois de distances entre nœuds et de bruit constant, il est évident qu'il existe un point de fonctionnement optimal, en termes de puissance d'émission, qui donne un minimum d'énergie consommée pour une qualité de services donnée. Malheureusement, en pratique l'environnement est rarement figé, et le Rapport Signal-sur-Bruit peut varier énormément, ce qui explique pourquoi la puissance d'émission reste fixée au maximum. Même si ce n'est pas une solution optimale du point de vue énergétique, cela permet d'assurer la qualité de services voulue de manière permanente.

La Fig. 16 représente l'énergie par bit transmis avec succès en fonction de différentes puissances d'émission. Chaque courbe est tracée pour un niveau de bruit donné et permet d'illustrer le point optimal de fonctionnement en termes de puissance d'émission. Pour des bruits faibles, il est possible de choisir une puissance d'émission relativement faible, contrairement au cas de bruit fort. A cause des retransmissions nécessaires, l'énergie nécessaire augmente de façon très abrupte à gauche du point optimal, et il faut donc faire très attention à ne pas se retrouver dans cette situation.

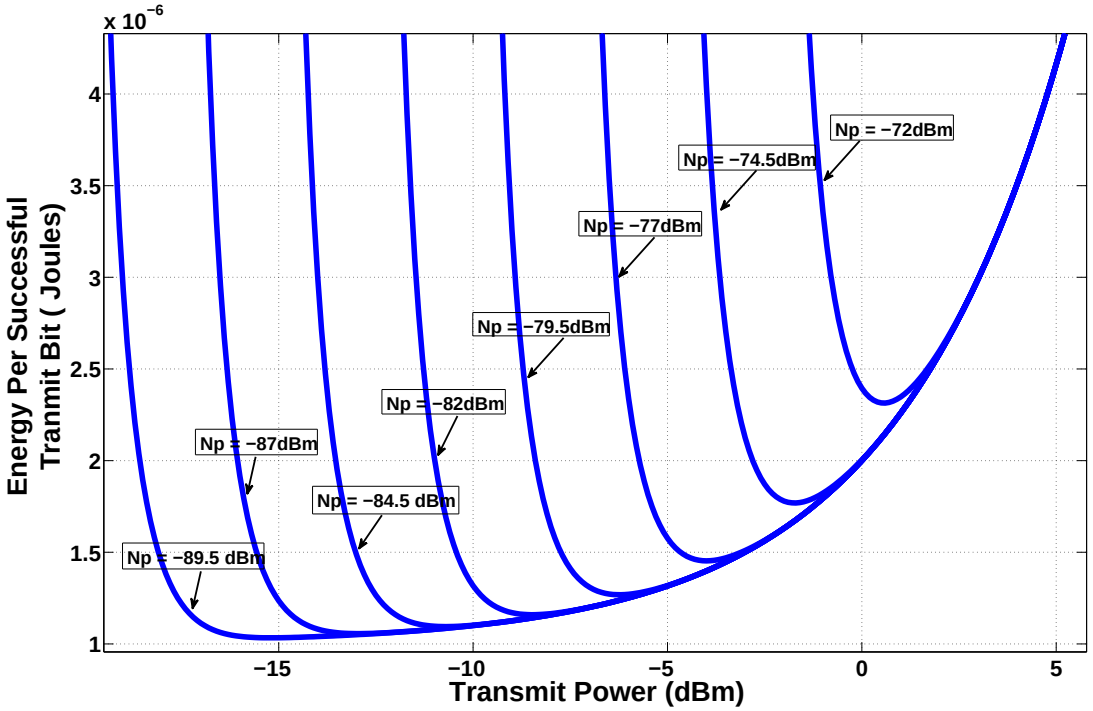


FIGURE 16: Energie consommée par bit transmis avec succès pour différents niveaux de bruit Np.

L'algorithme pour choisir le meilleur niveau de puissance d'émission doit être de faible complexité, pour être à la fois efficace en énergie et assez rapide pour que les conditions de propagation soient considérées comme constantes pendant assez longtemps pour que le coût de l'adaptation soit rentable.

Ce processus de sélection de puissance peut être décomposé selon les étapes suivantes

- Lorsqu'un paquet de données est reçu, cela donne une indication sur la puissance reçue. Durant les simulations, on considère que l'on connaît le niveau de bruit au niveau du récepteur, ce qui nous permet de connaître le RSB. En pratique, le niveau de bruit peut être estimé à partir du RSSI (Receive Signal Strength Information).
- A partir de cette estimation, il est possible de calculer le Taux d'Erreur Binaire.
- En considérant 10^{-3} comme seuil minimal de TEB (pour une modulation BPSK), c'est-à-dire en tenant compte des retransmissions éventuelles, on calcule l'énergie par bit E_b. Lorsque l'on parcourt les courbes d'énergie (comme celle illustrée par la Fig. 16) de la droite vers la gauche, c'est-à-dire en diminuant la puissance d'émission pour un niveau de bruit donné, l'énergie diminue jusqu'à un point de fonctionnement optimal avant de remonter très abruptement à cause des réémissions nécessaires.

Grâce au paquet d'accusé de réception, le nœud de réception informe l'émetteur de la puissance d'émission à utiliser pour le prochain paquet, considérant que le canal ne va pas ou peu varier dans ce laps de temps. En pratique, pour réduire le temps d'optimisation, on pourra utiliser des tables (LUT pour Look-Up Table) de correspondance entre le niveau de bruit et la puissance d'émission optimale.

Le transceiver radio *cc-2420* propose 8 différents niveaux de puissance d'émission, à savoir $0, -1, -3, -5, -7, -10, -15, -25$ dBm. En pratique, on ne peut donc pas choisir la puissance d'émission optimale, mais le niveau d'émission immédiatement au-dessus parmi les précédents. Pour les hauts RSB, de faibles puissance d'émission seront sélectionnées, tandis que pour un niveau de bruit élevé, la puissance d'émission va tendre vers 0dBm, c'est-à-dire que l'adaptation n'apportera pas de gain.

La Fig. 17-b montre la consommation d'énergie par bit transmis avec succès en fonction des variations temporelles du canal de propagation. Les trois courbes proposées

représentent l'énergie pour une puissance d'émission fixée à 0dBm, l'énergie dépensée pour une puissance optimale (théorique) d'émission, et l'énergie dépensée en adaptant la puissance d'émission aux niveaux permis par le *cc-2420*. Lorsque le bruit est faible, la technique adaptative suit les variations du canal et est toujours bien plus efficace. En revanche, lorsque le bruit est élevé, comme sur les pics entre 600 – 800s et 1600 – 1800s, le gain est nul.

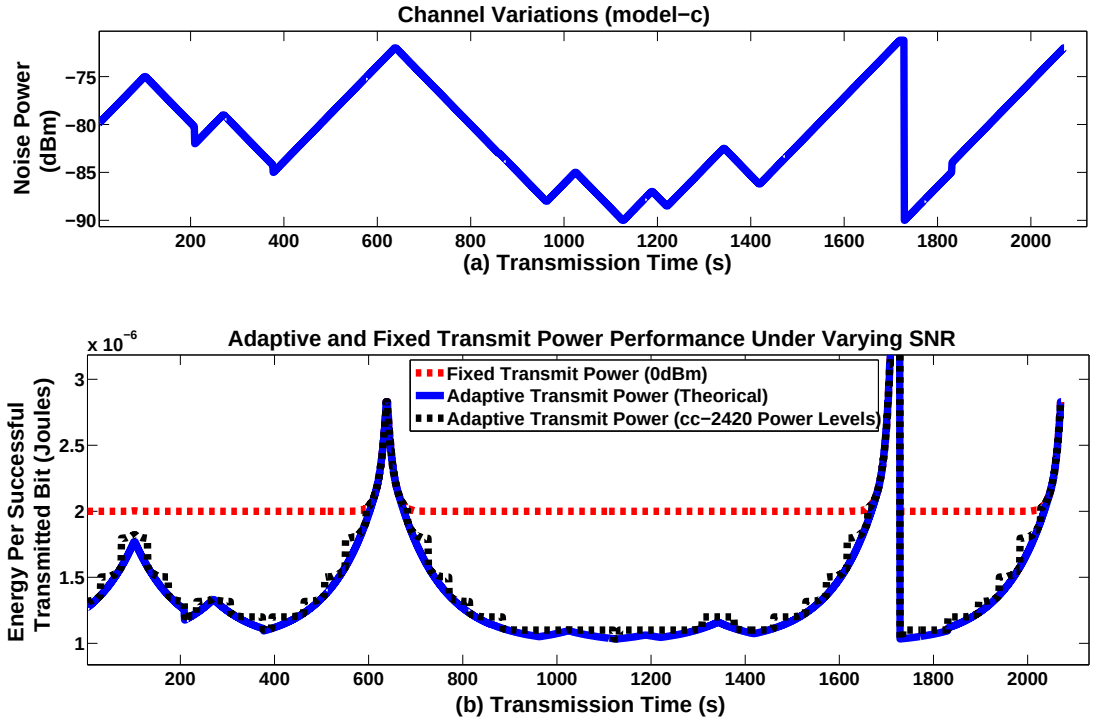


FIGURE 17: (a): Variations temporelles du canal (b): energie dépensée correspondante.

Nous venons de montrer nous pouvions adapter dynamiquement la puissance d'émission pour suivre les variations du canal. Dans les réseaux de capteurs, la plupart des communications basées sur le protocole 802.15.4 utilise une puissance fixe et maximale (0dBm). Nous allons donc comparer le gain de notre technique adaptative par rapport à cette puissance, comme illustré par la Fig. 18 qui représente l'énergie en fonction du niveau de bruit du canal en utilisant le même modèle de canal que précédemment. Cette figure démontre l'intérêt de l'adaptation, en particulier pour les faibles niveaux de bruit, pour lesquels utiliser une puissance maximale génère un gaspillage important. Lorsque le canal est trop mauvais, le transceiver radio ne peut s'adapter à ces conditions et l'adaptation est inutile.

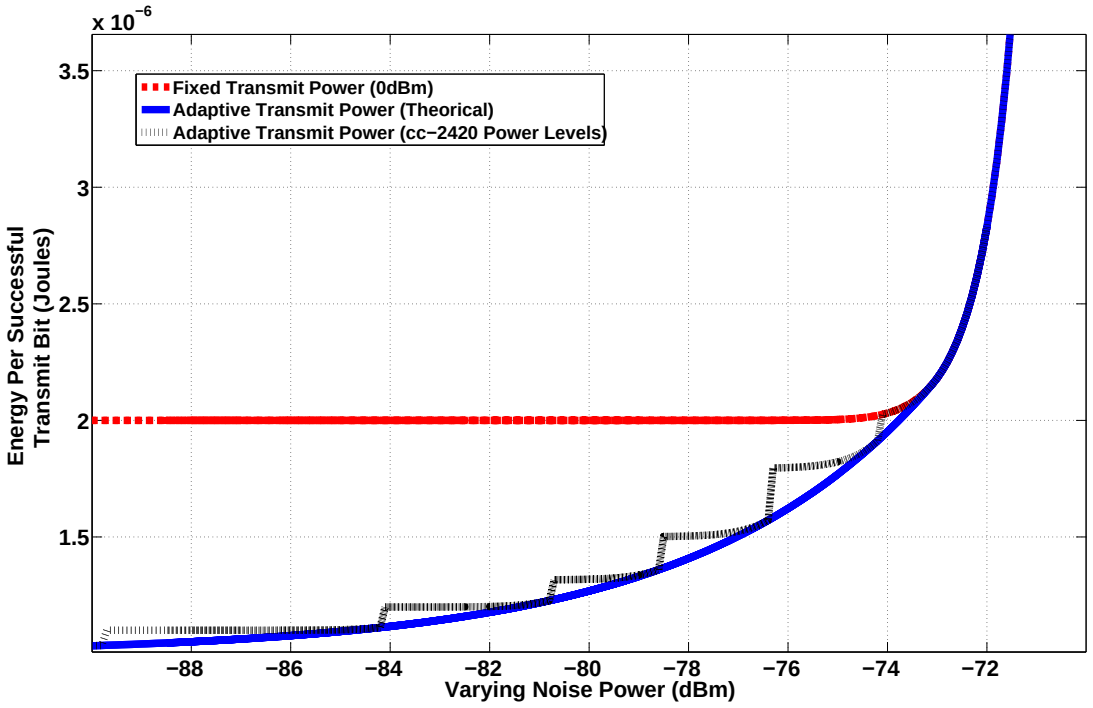


FIGURE 18: Comparison of energy performance between an adaptive transmit power optimization versus the fixed transmit power at maximum level of *cc-2420* i.e. (0 dBm).

0.6 Conclusions

La pression environnementale rend aujourd'hui le label énergétique crucial pour la plupart des industries, et le monde des télécommunications n'échappe pas à cette tendance, avec l'avènement de la green radio. La modélisation énergétique est un problème préalable à toute optimisation, et nous avons pris dans nos travaux le parti d'utiliser des mesures expérimentales sur la plate-forme PowWow pour concevoir un modèle hybride, dont nous avons démontré la précision au cours de cette thèse.

Grâce à ce modèle, nous avons pu caractériser les différentes phases d'une communication hertzienne, et observer les lieux de dépense énergétique importante et le potentiel d'optimisation. Nous avons ainsi tenté de réduire l'énergie nécessaire à une transmission, en jouant sur la couche physique pour essayer de diminuer la puissance radio en fonction des conditions de propagation. Mais nous nous sommes surtout attachés à réduire, au niveau de la couche d'accès, l'écoute passive inutile qui représente la partie prépondérante du gaspillage énergétique. Le protocole TAD-MAC, capable de s'adapter dynamiquement aux différents débits nécessaires dans un réseau corporel pour acheminer

l'information des différents capteurs physiologiques, permet ainsi une économie d'énergie très intéressante par rapport aux protocoles asynchrones existants.

Chapter 1

Introduction

Wireless Sensor Networks (WSNs) are a fast evolving technology having a number of potential applications in various domains of daily-life, such as structural-health and environmental monitoring, medicine, military surveillance, robotic explorations, etc. Advancements in Micro-Electro-Mechanical-Systems (MEMS) technology, wireless communications, and digital electronics have facilitated the development of low-cost, low-power, and multi-functional sensor nodes that are small in size and communicate efficiently over short distance. Thus the emerging field of WSN combines sensing, computation, and communication into a single tiny device (a WSN node).

A WSN consists in spatially distributed autonomous and embedded devices that cooperatively monitor physical or environmental conditions as shown in Fig. 1.1. The data (such as temperature, vibrations, sounds, movements, etc.) collected by each node is reported to a sink station in a hop-by-hop fashion using wireless transmission. Such data can then be processed and analyzed for a better understanding of the monitored environment. The need to perform these operations in a least intrusive fashion has motivated the development of small and low-power wireless devices. Engineering a WSN node device is a tough challenge, as the design must enforce many severe constraints among which energy consumption is often the most critical one due to the small size of a node and its difficult access after deployment. Since energy consumption remains one of the most important issue in WSN, this thesis focuses on algorithmic-level power optimization techniques for low-level layers of the communication protocol stack.

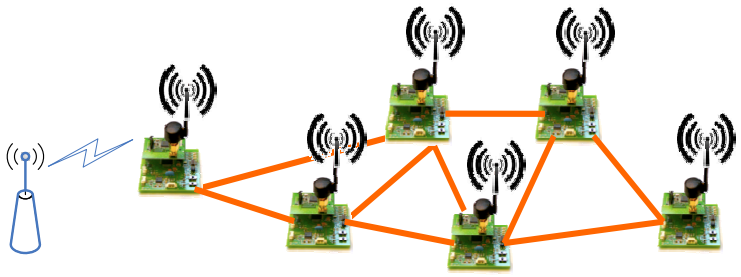


FIGURE 1.1: Sensor networks consisting of small nodes with an ability to sense the physical environment and communicate the information to the local (more powerful) base station.

In the last decade typical sensing (such as temperature, humidity, vibrations etc.) have been realized by using sensor networks and recently more applications (such as cognitive radio sensor networks, health-care, sports etc.) are emerging as the WSN technology is maturing. Though, there are still important challenges to be addressed. Limited battery capacity has put number of emerging applications on stand-by and energy efficiency remains one of the top challenges in WSN.

Reduction in energy consumption can be viewed from both software as well as hardware points of view. As far as the hardware component is concerned, radio transceivers integrated circuit have become extremely efficient in terms of power consumption. There are mainly two types of radio chips, one focused on the receive current consumption and keeps it low but the transmit current is on higher side. Whereas in the second type of radio chips designers try to keep the balances in between transmit and receive currents. Radio transceivers are optimized better in terms of energy consumption in comparison with previous decades but still need improvements (innovation in technologies). From the software point of view, protocol stacks not only within each layer but also through cross-layer techniques have achieved significant improvements for extending lifetime of the sensor networks but have not reached the limits. Moreover, as the impact of software controlling the hardware is vital, there is still room for a software part of the WSN node to perform better.

In the context of WSN, earliest research activities targeted the development of low-power devices allowing the development of large-scale networked sensor systems in the project SensIT [121]. In the same way there are number of research projects for sensor networks: SensorSim [89] resulted in a simulation framework for analyzing sensor networks, SmartDust [116] provided a millimeter scale platform using optical reflector technology and

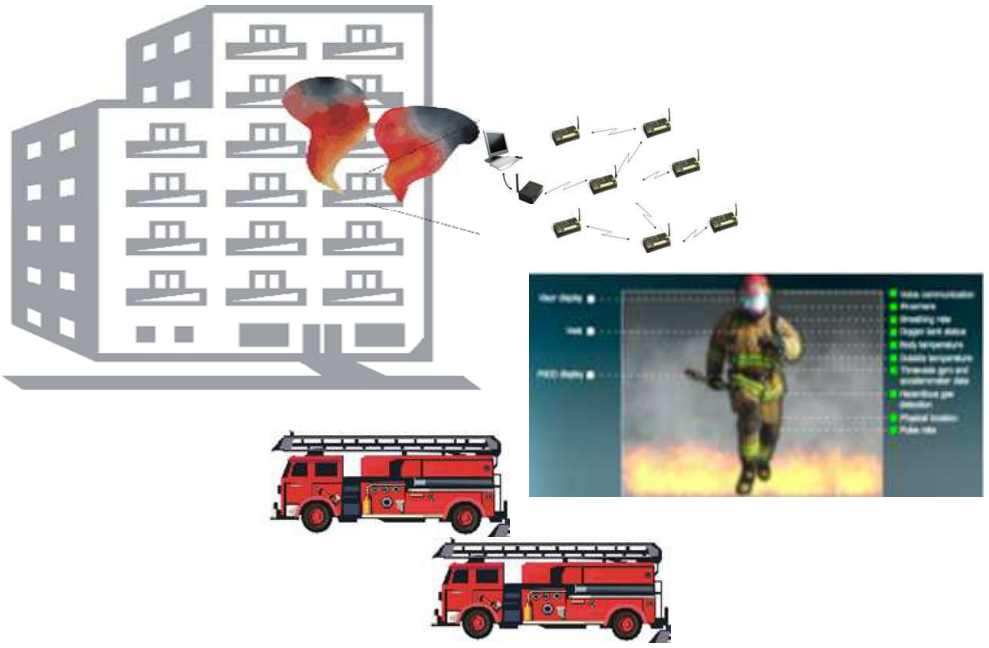


FIGURE 1.2: An application context of GEODES Project.

GEODES (Global Energy Optimization for Distributed Embedded Systems) [90] targets the energy optimization in disaster management applications as shown in Fig. 1.2. The work conducted in this thesis is carried out under the GEODES project at Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) Laboratory, from University of Rennes1. From the application point of view it has to perform two operations, first, to sense the environment parameters, mainly temperature and smoke second, to transmit bursts of data (such as images) of the affected area inside the building. The complete fire fighter scenario is depicted in Fig. 1.3.

The GEODES project provides design techniques, embedded software and accompanying tools needed to face the challenge of allowing long power-autonomy of connected embedded systems, which are becoming pervasive and whose usage is significantly rising [91]. It approaches this challenge by considering all system levels, and notably emphasizes the distributed system view focused with WSN.

1.1 Motivation and Objectives

Wireless sensor networks are the solution for remote monitoring and the most important design constraint is the power consumption. Energy harvesting (from many natural environmental resources) has provided an effective option to increase the lifetime of the

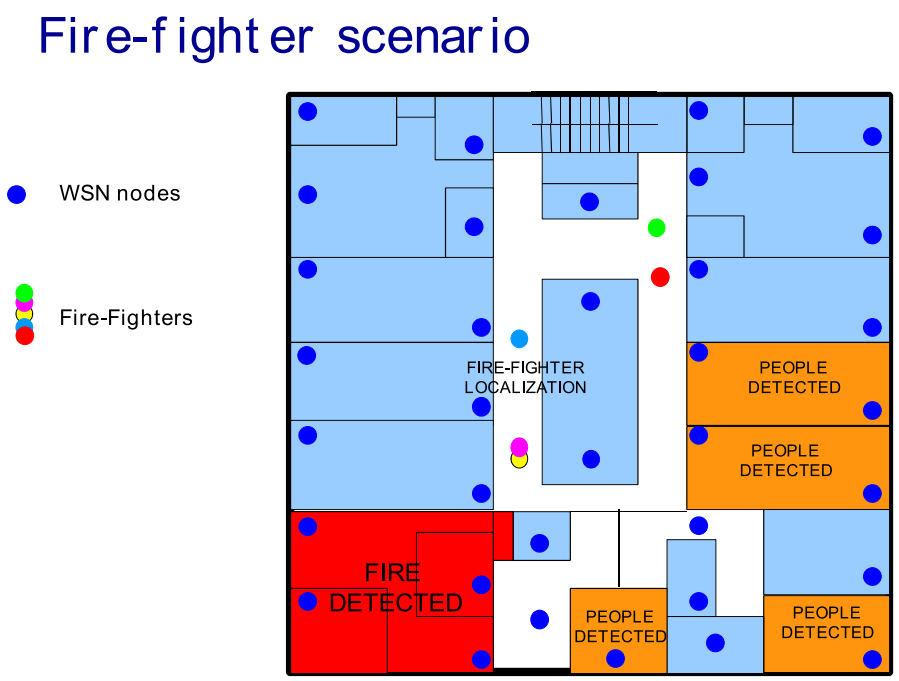


FIGURE 1.3: Fire fighter scenario within a building (GEODES Project).

sensor networks but it is not completely mature and in many cases not enough to be relied. Therefore the fundamental question about how to increase the lifetime of the sensor networks is still valid.

In these regards, our motivation is to create completely autonomous (self-organized) and distributed wireless sensor networks. To realize this motivation of having a self-organized system into practice there are two possibilities:

- **Emergent behavior** is a natural phenomenon where components interact on the micro-level and organize themselves (without having global knowledge) in such a way that leads to a macro organization. One of the example is ant-colony, a single ant has no global knowledge about the task it is performing, their actions are based on local decisions and are usually unpredictable. But, the intelligent behavior naturally emerges as a consequence of the self-organization.

- **Design strategy** is an adaptive design for self-organization in which the system components have to follow certain local rules (on the small-scale) intended to reach a desired global behavior (at the large-scale). The configuration has to converge with each step of a computation that depends upon interactions with all local and global elements. The final result inhibits such advanced complexity that it could never have been designed all at once.

In this thesis we propose some local-level optimizations that can lead towards a globally optimized system. The focus of these optimizations are at lower layers, i.e. physical (PHY) and MAC, with prime concern of achieving maximum energy efficiency. Therefore, the objective is to develop energy optimized techniques and solutions for the PHY and MAC layers under dynamic environment conditions.

1.2 Contributions

This work proposes power-aware adaptive techniques for WSN. Following are the three main contributions in this thesis.

- The first contribution is a hybrid energy model for accurate energy analysis for WSN nodes. First of all, it is important to identify which are the key factors of energy consumption at the lower layers in WSN before developing power-aware optimized techniques for them. Therefore, it is important to know energy consumption details of WSN nodes. There are two aspects in the proposed energy model. First, the model is based on different communication scenarios that can occur in distributed asynchronous sensor networks. One of the example from these scenarios is *data-collision*; at one instant, two nodes can communicate each other without any errors or collisions but the same nodes can have collisions when they communicate at another instant. Second, the analysis relies on analytical model and power measurements and, with the combination of both we achieve an integrated model which computes the cost of a communication of the data packet. The accurate power measurements provide the details of the current level in all different scenarios that were identified. The measurements do not assume constant current level (as is the case in most of the state-based measurements) during the communication. For example, while performing channel sensing the current level is not the same as that of the receive state. Therefore the scenarios and accurate power measurements help to propose a pragmatic energy model.

- After providing the energy consumption of the hardware and software components through scenarios, it is concluded that there is a significant amount of energy waste due to non-optimized MAC layer. This energy waste includes overhearing, overheads, collisions and most importantly idle listening and it is suggested therefore,

that a MAC protocol which can adapt the wake-up interval dynamically can reduce $10 - to - 100$ times the energy consumption if idle listening is minimized. So, the second contribution is the design and development of the traffic-aware dynamic (TAD) MAC protocol. The protocol is based on a simple yet effective adaptive algorithm which utilizes a traffic status register to compute the precise wake-up interval. The algorithm is designed in such a way that it can adapt itself to different traffic variations. TAD-MAC proposes an on-line solution, which means it adapts the wake-up interval through a gradual convergence process and achieves a coherent pattern of traffic status register that ensures the precise wake-up interval. TAD-MAC has been thoroughly analyzed to ensure that the algorithm always converges to a steady state and to any traffic variations that are possible in WSN. In this regard, the parameters of the adaptive algorithm are characterized and analyzed as a control system (in open-loop and closed-loop forms) to understand the behavior of the adaptive algorithm during run time with different variations from the environment. TAD-MAC protocol is proved to be extremely efficient in the context of wireless body area sensor networks (WBASN) both in terms of energy efficiency as well as latency as it outperforms the other low-power state of the art protocols. TAD-MAC improves the lifetime of WBASN by a factor of 3 to 6 times.

- After achieving most of the optimizations at the MAC-layer, the next contribution of this thesis is on transmit power optimization at the physical layer. In this regard, an adaptive transmit power optimization technique is applied under dynamic channel variations to reduce the energy per successful transmitted bit at the physical layer. The output power is adaptively tuned to optimal power level on a link-by-link basis. Each node locally (for all the neighbor nodes) adapts the power according the signal-to-noise ratio (SNR) variations. The optimization is achieved under slow varying channel at the reception of a single packet. There exists an optimal point as the energy per bit reduces until a certain transmit power level and after that it starts increasing (as the signal power is too weak) due to retransmissions. So, these optimal points are identified under varying SNR conditions which provides some energy saving at the transmitter end. Different radio transceiver power profiles are used to show the gain over fixed transmit power. It is found that by dynamically adapting the transmit power, it can help to reduce the energy consumption by a factor of 2. Although our contribution is more a

proof-of-concept than an algorithm, some implementations issues are investigated for WSN platforms.

1.3 Thesis Organization

After this introduction chapter, the rest of the thesis is organized as follows.

Chapter 1 introduces state-of-the-art which is mainly focused on three aspects of WSN. Following a brief introduction of few applications and cross-layer design, preamble sampling class of MAC protocols is explained. Then, various energy models which are commonly used in WSN are presented and finally some features of the existing WSN platforms and network simulators are presented.

Chapter 2 describes our first contribution in the form of energy modeling. After the introduction and explanation of the related work on energy consumption modeling, analytical models of the low level layers such as data link, MAC and physical layers are presented. The protocol stack implementation and power measurements on a hardware/software platform (PowWow) are explained. Further a scenario-based hybrid energy consumption model that takes into account the analytical models as well as the real measurements is proposed. Finally, the model is validated under different test cases and some of the applications of the proposed energy model are presented.

Chapter 3 presents the traffic-aware dynamic MAC (TAD-MAC) protocol. After an introduction describing same energy efficient MAC protocols and their related work, the design principle of proposed TAD protocol is presented. An adaptive system characterization as a control system allows to understand the convergence behavior of the the proposed TAD-MAC is described. Finally, implementation in a network simulator and performance analysis are presented.

After developing an understanding of the TAD-MAC algorithm, the protocol is exploited in the context of wireless body area sensor networks (WBASN). Chapter 4 presents the application of TAD-MAC for WBASN and starts by the context of WBASN followed by related work. Then, the proof-of-concept and simulation results are presented, further TAD-MAC is compared with other existing energy efficient MAC protocols under fixed and variable traffic for various widely used radio transceivers chips.

Chapter 5 introduces adaptive transmit power control techniques. After a brief introduction and some related work, the adaptive transmit power control technique is presented. Finally some implementation considerations are discussed to propose an algorithm for the future work.

Finally, the global conclusion of the thesis along with future perspective are presented.

Chapter 2

Wireless Sensor Networks

Wireless sensor networks (WSN) consists of small sensor nodes with an ability to sense the physical environment and communicate the information to each other and to a base station. Wireless sensor networks are based on tiny, low-cost and limited power devices (nodes) that have the capability to sense the environment. These devices communicate with each other to transport the information over short distances. Typically a sensor node consists of a sensing unit (sensor/actuator), a processing unit (microprocessor), a transceiver (radio chip) and a battery unit. WSNs have some key characteristics different than ad-hoc networks such as low traffic, small network, fast deployment, severe power constraints, short radio range and limited mobility.

There are several applications of WSN ranging from health care, clinical monitoring, border surveillance, environmental control, natural disasters, to battle fields. Some of these applications are listed below:

- Surveillance applications are used for intrusion detection, target tracking, video monitoring for security purposes. Sensor nodes send burst of data as soon as an event is occurring. These applications require the sensor nodes to have high computational power and video capabilities. [102] illustrated one of the example being used to monitor a farm house.
- In disaster management applications such as fire eruption, tsunami, earthquake, etc. the nodes can be deployed in forest, buildings, near the sea shores to periodically transfer the information to the monitoring station. In such applications

most of the sensor devices are powered by limited batteries and energy efficiency is very important. Such an application is explained in [74], where a sensor network quickly alerts the first responder for disaster management.

- Health-care and patient monitoring is a life saving application of WSN [52]. In the past few years with the help of technology (such as body sensor networks) there was a significant improvement both in terms of quickly reaching to the main cause of patient illness as well as in terms of comfortability of the patient [86]. Energy and latency are the important design constraints in these applications, moreover interference is an important issue.

- There are applications of WSN in underground as well, a sensor network could be deployed in mines [11] to collect various parameters such as hazardous gas level for the security and health safety of miners. Acoustics sensor networks, monitoring of sea habitats and soil monitoring are various important classes of underground applications [29], [79].

- Maritime applications take advantage of WSN for water distribution monitoring systems. [83] explains an application that targets an accurate channel design for better performance of distribution management system.

2.1 Reduced Open Systems Interconnection (OSI) model

Typically in WSN a reduced Open Systems Interconnection (OSI) model is utilized as shown in Fig. 2.1. It starts from the application layer, which makes sure that the data packet is reliably transmitted (up-link) or received (down-link) to or from the network layer with specific transmission rate. The network layer performs one of the important task in multi-hop environment to deliver the data packet to the correct node through the best possible route, it adds a routing header which includes the network address of the destination node in up-link transmission, whereas, in the case of down-link routing layer makes sure that the packet is forwarded to the correct destination node (if it is not the final destination).

Further, the packet is transmitted towards the data link layer (DLL) which is often integrated with the medium access control (MAC) layer. One of the important purpose

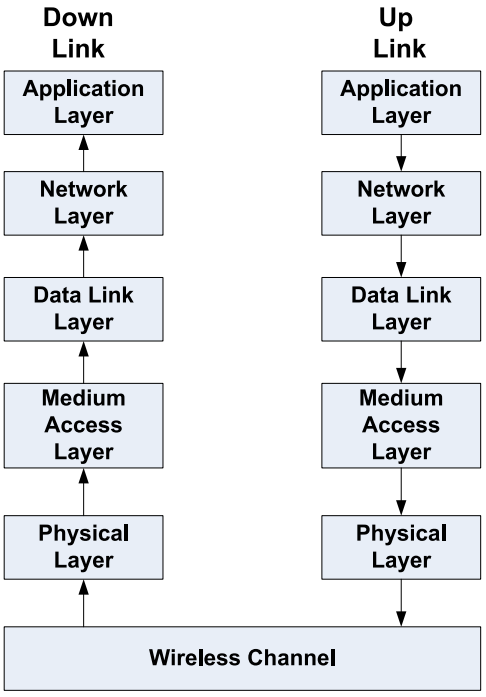


FIGURE 2.1: Reduced OSI model of wireless sensor networks for transmission (up link) and reception (down link).

of DLL is to add an error correcting code which is decoded in the down-link and the MAC layer appends its header inside the data packet and controls the access of the medium. Normally different types of packets are used (such as control and data) during the communication and it is the MAC layer which distinguishes between them. Therefore, if a node is waiting for a data packet and it receives a control packet so it is the role of MAC layer to ignore this control packet instead of forwarding it to the upper layers in down-link process. Finally the physical (PHY) layer appends its header and tunes various parameters that make sure an acceptable performance for packet transmission and reception. Furthermore, it is the interface to the outside environment for a sensor node.

2.2 Medium Access Control

Medium Access Control (MAC) is the ability of a node to efficiently share the wireless medium with the other nodes in the network. In the design of energy aware MAC protocols, the main causes of energy consumption are idle listening, overheads, overhearing and collisions. Therefore, in order to achieve energy efficiency, these factors need to be minimized, but there exists a trade-off for the optimal design. For example, the protocol

with the aim to reduce idle monitoring and collisions demands extra synchronizations and overheads, whereas, reducing the overheads and synchronizations results in an increase in energy waste due to collisions.

In the context of energy efficient WSN protocols, the preamble sampling method is an attractive option for light-traffic WSN [19]. The general mechanism of a preamble sampling protocol is shown in Fig. 2.2. As the initialization of communication between sensors can be initiated by either a transmit or a receive node (depending upon the specific protocol being used), both cases are depicted in Fig. 2.2. If the communication is initiated by the receiver then the receive node will send the preamble to the transmit node and the transmit node will respond with the data packet, whereas if it is initiated by the transmitter then the preamble will be sent by the transmit node followed by the data packet. The receive node wakes up periodically at a sampling interval to sense the channel activity and if it does not find any preamble it goes to sleep mode immediately. It is to be noted that, preambles have to be long enough such that the intending receive/transmit node can be able to receive the preamble on the wake up and further to keep the radio on for receiving the subsequent data packet.

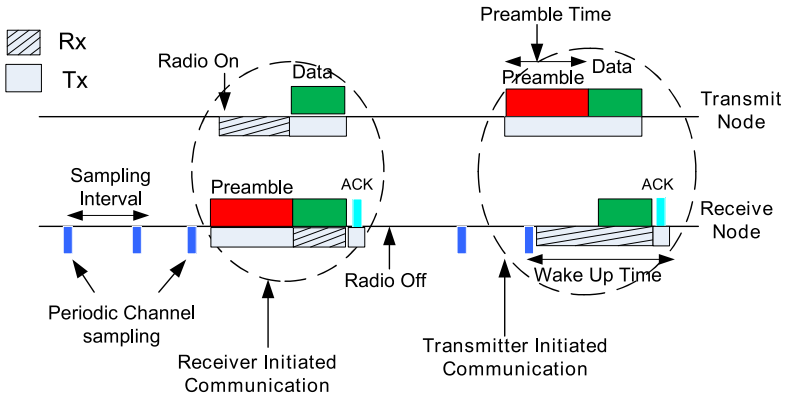


FIGURE 2.2: General mechanism of preamble sampling protocols. The communication can either be initiated by receive or transmit node. If the communication is initiated by the receiver then the receive node will send the preamble to the transmit node and the transmit node will respond with the data packet, whereas if it is initiated by the transmitter then the preamble will be sent by the transmit node followed by the data.

Preamble sampling MAC protocols are based on a non-scheduled mechanism without any synchronization among the nodes, which means that each node is completely independent of its own active/sleep strategy. The Preamble sampling category includes energy efficient MAC protocols such as Cycled Receiver, i.e. RICER (Receiver Initiated Cycled Receiver) [82] or TICER (Transmitter Initiated Cycled Receiver) [82], LPL (Low Power Listening) [95], BMAC (Berkeley MAC) [95], XMAC [24], WiseMAC [60], etc. These

protocols reduce the cost of extra overheads (in comparison with scheduled based protocols) and synchronizations by having single or multiple preambles. In addition a protocol such as WiseMAC can adjust the duty cycle efficiently based on the 'wake up time' of the neighbor nodes, which results in a great reduction of idle monitoring, preamble size and probability of collisions. A brief description of some of these protocols is proposed below. RICER, which is mainly used in this thesis, will be presented in Chapter 3.

2.2.1 Berkeley-MAC Protocol

BMAC is a carrier sense multiple access (CSMA) based technique that utilizes low power listening and an extended preamble to achieve low power communication. Nodes have awake and sleep periods and each node can have an independent schedule. If a node wishes to transmit, it precedes the data packet with a preamble that is slightly longer than the sleep period of the receiver as shown in Fig. 2.3. During the awake period, a node samples the medium and if a preamble is detected it remains awake to receive the data. With the extended preamble, a sender is assured that at some point during the preamble the receiver will wake up, detect the preamble, and remain awake in order to receive the data.

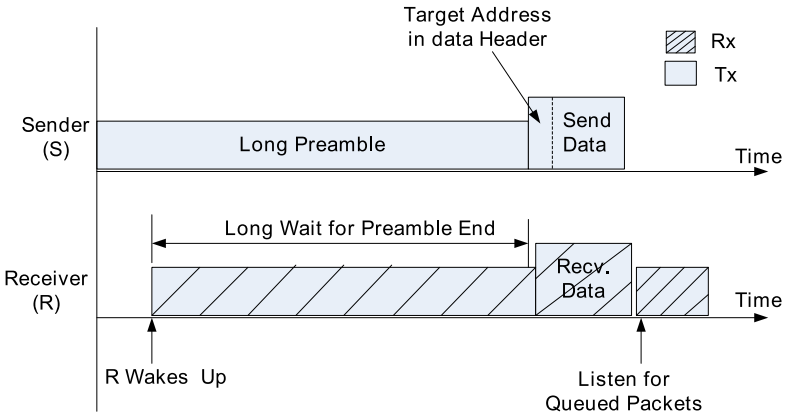


FIGURE 2.3: BMAC protocol.

BMAC improves the quality of clear channel assessment (CCA) by replacing a typical thresholding sensing mechanism with outliers identification technique. In this technique, outliers are found (much below the noise floor level) when the channel is free, whereas if the channel is occupied then the node on sampling the channel does not find any outliers and after five samples it backs-off for a certain duration. Significant advantage has been

reported by authors (in [95]) by reducing collisions and effective utilization of the channel access as outliers detection substantially outperforms thresholding.

BMAC is one of the pioneers of preamble sampling category of MAC protocols though, the utilization of long preamble results in extra energy consumption both at transmit and receive sides. This specific issue is optimized by reduced preamble XMAC protocol which is presented in next section.

2.2.2 Reduced Preamble XMAC Protocol

XMAC protocol [24] uses a series of short preamble packets to reduce energy consumption in comparison with BMAC. Fig. 2.4 illustrates the basic operation of the XMAC protocol. The destination address is embedded in the preamble packets, which helps to reduce a significant amount of overhearing at non-intended receive nodes. The ACK packet is used just after the reception of first preamble packet that hits the target receivers active period. It helps to reduce the preamble packets transmission as the data packet is transmitted immediately from the transmitter after receiving the ACK packet. Also, the size of the preamble packet is very short with redundant transmission of the same packet until the sender gets the ACK packet. Like in most of the preamble sampling MAC protocols, CCA is performed before the preamble packet is transmitted. After the data packet is received, the receiver waits for a short period to give a chance to any node that wants to send packets as can be seen from Fig. 2.4.

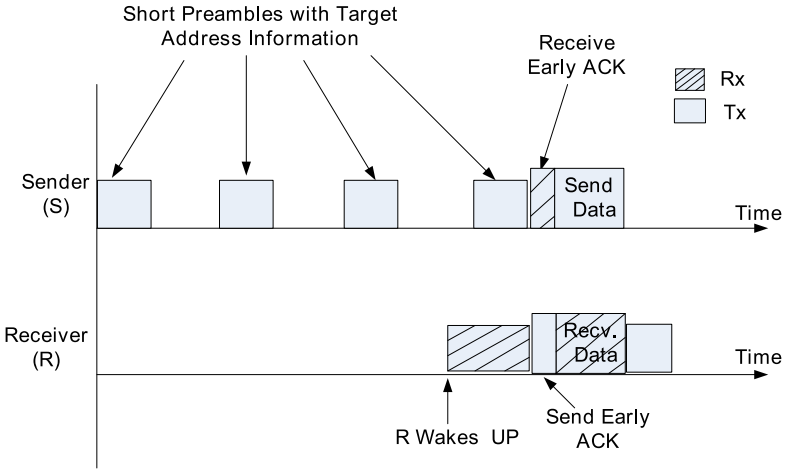


FIGURE 2.4: X-MAC protocol.

The X-MAC protocol provides more energy efficiency and lower latency operation by reducing the transmission energy and transmission period burdens, idle listening at the

intended receiver and overhearing by the neighboring nodes. One concern is that the gaps between the series of preamble packet transmission can be mistakenly understood by the other contending nodes as an idle channel, and they would start to transmit their own preamble packets which can lead to collision. One solution is to ensure that the length of the gaps must be upper bounded by the length of the listening interval [10].

2.2.3 WiseMAC protocol

WiseMAC [60] is also based on preamble sampling to achieve low power communications in sensor networks. Fig. 2.5 shows the principle of WiseMAC protocol. It uses a similar technique as that of B-MAC, but the sender learns the schedules of the receiver awake periods and schedules its transmission so as to reduce the length of the extended preamble. To achieve this, the receive node puts the time left for its next wake-up in the data acknowledgment frame. The next time the sender wants to send to that receiver, it can begin the preamble only a short time before the receiver will awake, taking into account possible clock drift. This reduces the energy consumption for sending much shorter preambles in comparison with BMAC preamble size.

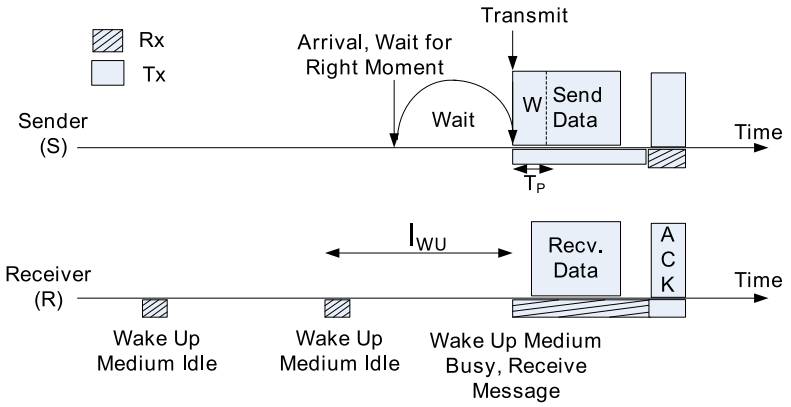


FIGURE 2.5: WiseMAC protocol.

WiseMAC pays special attention towards adaptively controlling the length of the preamble based on the frequency of communication. If the communication interval is shorter (i.e. high traffic), then the preamble length will be shorter, whereas, if the communication interval is longer (i.e. low traffic), then the preamble length can increase up to the wake-up interval of the receive node. This control of the preamble length (T_p) is obtained through $T_p = min(4\theta l, I_{WU})$, which is the minimum of $4\theta l$ and I_{WU} where θ is

the clock drift, l is the time elapsed since the last communication between the two nodes and I_{WU} is the wake-up interval of the receive node.

In addition, for low traffic loads where the preamble is longer than the data frame size, WiseMAC repeats the data frame in place of the extended preamble. Receivers process this data frame and if the node is not the intended recipient it returns to sleep. If the node is the recipient, it remains awake until the end of the transmission and sends an acknowledgment. While WiseMAC solves many of the problems associated with low power communications, it does not provide a mechanism by which nodes can adapt to changing traffic patterns (a more detailed explanation can be found in Sec. 5.4.2).

2.2.4 Cross-Layer Optimizations

It is widely known that a cross-layer design approach is necessary to achieve the highest life time and the best performance in wireless networks. As far as wireless sensor networks (WSN) is concerned, cross-layer optimization is an essential requirement and the challenge for the power constraint WSN is: how to tune the cross layers adequately? On the one hand, it is considered as the best solution to increase performance and life time of the sensor node but, on the other hand, the cross layer approach can significantly increase the complexity with the huge design space. One of the solutions is to interact between the layers based on the most influential parameters of each layer that have direct and indirect impacts, as it is the case in [107] and [131], and, at the same time, each layer needs to be optimized itself according to some constraints (energy, latency, bandwidth efficiency, etc). In these regards, the cross-layer parameters that can be characterized and interacted are as follows:

- The link layer contains the information about the channel behavior based on, for example, signal-to-interference noise ratio ($SINR$), bit error rate (BER), or *data rate*. So, the corresponding design parameters such as the selection of the suitable forward error correction codes (FEC), can be exchanged with the (MAC) and network layer.

- The MAC layer can support variable traffic rates [66] [61] in accordance with the application demand and based on the information exchanged from the link layer. This parameter of adaptive traffic rate can further be exchanged with the network

layer to adjust the routing accordingly to minimize the congestion in the network flow.

- Adjustments in the link/MAC layer have a direct impact on the output power at the hardware platform e.g., more robust error correcting codes may require more power. Similarly, a higher traffic rate requires more rendez-vous and higher order modulation schemes, which means more power consumption in the physical layer. Furthermore, the physical layer can share the received signal strength indicator ($RSSI$), and the link quality indicator (LQI) among MAC and link layers.

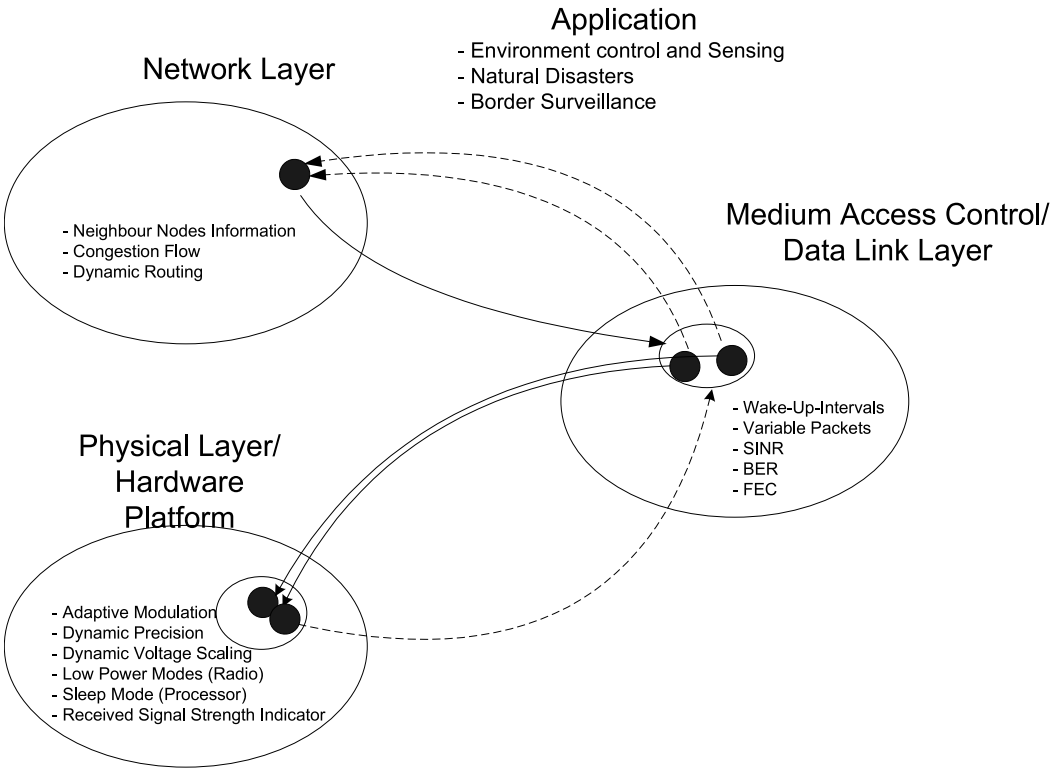


FIGURE 2.6: Generic framework: the parametric-based cross layer interaction between the network, link/medium access control and the physical layers for the given application. Performance metrics such as $SINR$, BER, $RSSI$, etc., can help to adjust different parameters such as FEC selection, *Wake up Intervals*, *Adaptive Modulation*, *Dynamic Precision* etc.

Ideally, a cross-layer optimization should be from the top to the bottom i.e., from application to physical layer respectively, but the actual design space gets very huge thus leading to a very complex problem. Therefore, different partial cross-layer design solutions such as in [131], [124], and [34], have been proposed with limited interaction between data-link, network and physical layers. This thesis also adapts this point of view and focuses on low layers such as DLL/MAC and PHY.

2.3 Energy Modeling

Energy modeling is one of the important aspects to accurately specify the energy consumption of the sensor node. There are number of models exist in the literature. Most of the models are state-based, other are components-based whereas, some are based on specific parameters of different layers of the protocol stack that have the dominating influence on energy consumption. In this section we will briefly explain few of the widely used energy models for wireless sensor networks.

2.3.1 Parameter-based Power Model

A wireless sensor node has a limited lifetime due to limited battery energy, mostly dissipated during the transmission and reception. The main reason of transmission and reception is either when a node generates a packet for a particular destination, or it has to receive and forward the packet towards the destination or it receives a packet when it is in idle state and the packet is not destined towards it. The power consumption model that is described below is a function of two parameters i.e. *packet size* and *distance* and is expressed as [13]

$$E_c = E_{tx}(packetsize, distance) + E_{rx}(packetsize) \tag{2.1}$$

where E_c, the total energy, depends upon the energy consumption of transmission E_{tx} and reception E_{rx}. The transmission energy consumption depends upon packet size as well as the distance between the nodes whereas the reception energy only depends upon packet size. *NS-2* is a network simulator (explained in Sec. 2.4.2) which utilizes the above power model with an extension which includes two threshold parameters (i.e. carrier sense and reception) that are used to precise the energy consumption of receiver. For packet reception there are three cases:

- If the distance between the sender and receiver is less than the reception threshold, the packet will be received correctly. In this case, energy will be consumed by the receiver for the packet reception.
- If the distance is greater than the reception threshold but less than the carrier sense threshold then the packet will be received with errors. In this case the energy will

be consumed for the packet reception by the receiver. Moreover the packet has to be re-transmitted.

- If the distance is greater than the carrier sense threshold then the packet will be lost, in this case the energy will not be consumed for the reception of the packet.

The *NS-2* energy consumption model technique is further verified in [35].

2.3.2 State-based Power Model

In order to evaluate the power consumption of different rendezvous schemes, a power model, known as MAC And Physical LAyer Power (MAPLAP) model, for wireless sensor nodes is presented in [80] and illustrated in Fig. 2.7-a. MAPLAP takes inputs from the physical layer, as well as the upper layers, to generate realistic evaluation of the power consumption per node. More in detail, a node is modeled as a probabilistic state machine with five operational states as shown in Fig. 2.7-b. These states are transmit (TX), receive (RX), acquire (AQ), monitor (MN) and sleep (SP) states. A node transmits and receives packets in the transmit and receive state, respectively. The acquire state is different from the receive state in that it represents the computation and power intensive timing, phase and/or frequency acquisition phase in which a node attempts to synchronize bits in the packet by receiving the preamble. Therefore, the acquire state always precedes the receive state. A node carrier senses the channel in the monitor state, awaiting for incoming packets. Finally, in the sleep state, most of the components in the node, including the transceiver circuitry, are powered off and the node consumes minimal power. A node transits among these five states over time.

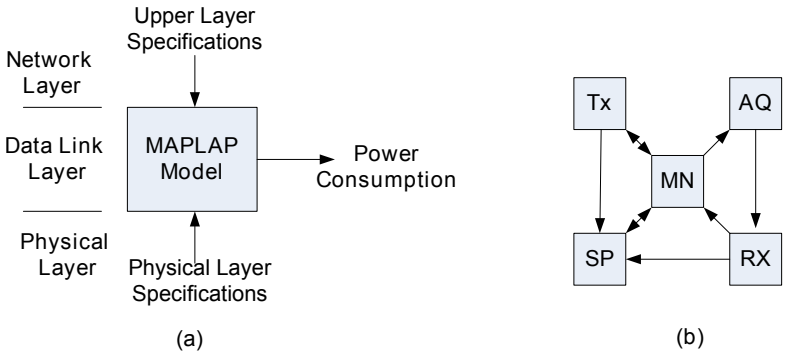


FIGURE 2.7: (a) MAC And Physical LAyer Power (MAPLAP) Model. (b) State-based wireless sensor networks.

The transition from some of the states to some other states consumes a non-zero turnaround power, P_{turn}. However, since the transition usually takes very less time, nodes are assumed to be in exactly one of the five states at any given time instant. Given this setup, the average power consumption per node can be expressed as

$$P_{tot} = \delta_{tx}P_{tx} + \delta_{rx}P_{rx} + \delta_{aq}P_{aq} + \delta_{mn}P_{mn} + \delta_{sp}P_{sp} + \sum P_{turn} \tag{2.2}$$

where δ_{tx}, δ_{rx}, δ_{aq}, δ_{mn} and δ_{sp} are the average percentage of time (or probability) of being in each state, and P_{tx}, P_{rx}, P_{aq}, P_{mn} and P_{sp} are the power consumption levels in each state. $\sum P_{turn}$ accounts for all non-zero terms of turnaround power. The δs are functions of the rendezvous schemes under analysis, as well as scenario variables such as traffic load and network density. The P variables are obtained by analysis and/or measurement of actual transceivers, whereas more details of deriving these expressions can be found in [80].

2.3.3 Component-based Power Model

[99] presented a component based power model as shown in Fig. 2.8 that illustrates the internal structure of a communication module found in a typical WSN node, and defines the power consumption of each component.

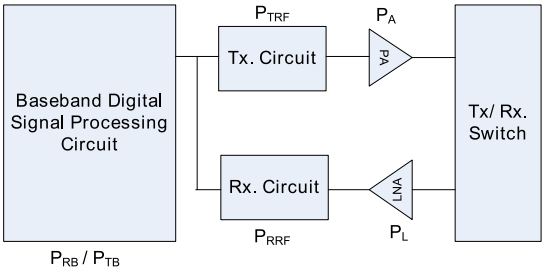


FIGURE 2.8: Component-based power model for WSN.

Based on the structure and power consumption of each component, the total power consumption for transmitting and for receiving, denoted by P_T and P_R, are expressed as

$$P_T(d) = P_{TB} + P_{TRF} + P_A(d) = P_{T0} + P_A(d) \tag{2.3}$$

$$P_R = P_{RB} + P_{RRF} + P_L = P_{R0} \tag{2.4}$$

where $P_A(d)$ is the power consumption of the power amplifier with the transmission range d is a function of it. Since P_{TB} and P_{TRF} do not depend on the transmission range, the two components can be modeled as a constant, P_{T0}. Similarly, the power consumption of the receiving circuitry can be modeled as a constant, P_{R0}, since P_{RB} and P_{RRF} are clearly not dependent on the transmission range, and P_L is also a constant while assuming that the LNA is properly designed and biased to provide the necessary sensitivity to reliably receive, demodulate and decode a minimum power signal, P_{Rx-min}.

Another component-based model is presented in [40], this model estimates the energy consumption of the components through on-line software-based mechanism. The sensor nodes frequently switch on and off their components (such as the communication device, LEDs, or sensors) to conserve energy. The energy estimation mechanism is invoked every time a hardware component is switched on or off. When a component is switched on, the estimation mechanism stores a time stamp. As the component is switched off again, a time difference is produced and added to the total time that the component has been turned on. The estimation mechanism keeps a list of all components and the time of which they have been turned on. The mechanism then uses the current draw of each component to produce an estimate of the total energy consumption.

2.4 Hardware Platforms and Network Simulators

Fig. 2.9 shows the fundamental components of a WSN node [68]. The sensor (or actuator) is the interface to the physical world, which observes (or control) the physical parameters of the environment. A controller (processor and co-processor) is used to process all the relevant data, whereas the purpose of communication device (such as radio) is to communicate the information over wireless channels. The memory block for the storage of data and program is used. Finally a limited capacity battery is used and a DC/DC converter is often required to scale the voltage either up or down to meet the requirements of the processing components.

2.4.1 Sensor Platforms

Sensor devices typically use batteries such as rechargeable AA batteries with voltage range from 2.7V to 3.3V. In most of the WSN applications and in deployed networks

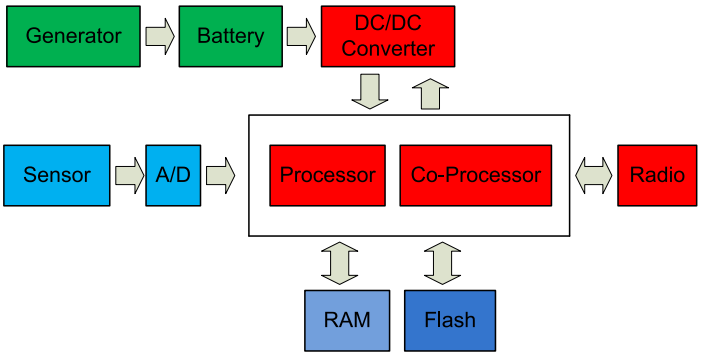


FIGURE 2.9: Basic Block Diagram of Wireless Sensor Node.

it is often difficult to replace these batteries therefore it is necessary to have energy efficient design approaches at the system level as well as at the network level. Some of the important features of available WSN platforms are listed in Tab. 2.1 and these motes are briefly explained below.

- **Stargate** [1]: This WSN mote has 32MB flash memory and a maximum clock speed of 400MHz that allows its utilization for image processing or video surveillance applications.
- **Imote** [1]: It has an operating frequency range between 13MHz and 416MHz and the energy can be saved for the applications that do not have stringent real-time requirements. However, it can also meet the real-time deadlines for high speed applications.
- **Mica2/Micaz** [1]: These devices have low operating frequencies and are typically used for applications that have low traffic and can not be used for high computing multimedia applications.
- **TelosB** [1]: It also operates at lower frequency and that is why it is used for typical sensing applications.
- **PowWow** [63]: PowWow like other state-of-the-art sensor devices is based on $MSP430$ microprocessor, the $cc2420$ radio chip, but it also provides hardware accelerator capabilities which is unique among other widely used sensor motes by adding a low-power FPGA. Further it is based on modular-basis with one base card and on top of it other processing cards, which makes it easier to upgrade.
- **SHIMMER** [108]: Shimmer is a small wireless sensor platform that can record and transmit physiological and kinematic data in real-time. It is mainly designed for

health applications and used as a wearable sensor. Shimmer incorporates wireless ECG, EMG, GSR, Accelerometer, Gyro, Mag, GPS, Tilt and Vibration sensors. It has up to 2GB flash memory with rechargeable lithium polymar (LiPo) battery with capacity nearly 250mAh.

- **WiSeNet** [43], [38]: WiSeNet is a low power system on chip which operates in sub-giga bands such as 433MHz ISM and 868MHz SRD (short-range-device) and has some power management features. WiseNet platform includes couple of analog-to-digital converters, a control unit based on the Cool-RISC microprocessor with on-chip low-leakage memory.

- **Tyndall/UCC Sensor Node** [92], [56]: Tyndall National Institute and University College Cork's sensor node is based on two complementary (i.e. 25mm by 25mm and 10mm by 10mm) heterogeneous wireless sensor nodes. 25mm node (operating in 433MHz, 868MHz and 915MHz frequency bands) contains Atmel ATmega128 micro-controller and Nordic nRF905 radio transceiver along with co-processor FPGA as main processing components. Whereas 10mm node is a single integrated (radio (nRF905) and micro-controller (8051)) chip nRF9E5 with 4KByte of program memory and EEPROM for non-volatile program storage.

2.4.2 Network Simulators

The impact of designing a new MAC or routing protocol, physical layer optimizations or low-power strategies, need to be evaluated in the context of the global network performance. This performance evaluation through the development of sensor networks is a complex and time consuming process. Thus, simulating tools for sensor networks are generally considered for evaluating new sensor network related approaches. Some simulators can be used for measuring the performance of a routing protocol with respect to shortest path finding mechanism while few of them focus on power efficiency and energy reduction in the MAC protocol. Most of them allow the development of new protocols and algorithms, and by following certain rules, these protocols can be integrated with built-in modules and can be easily tested and verified. Numerous WSN simulators can be found in the literature, and a brief explanation on few of these simulators is given below.

TABLE 2.1: Wireless Sensor Platforms.

Devices	**Processor**	**Memory**	**Wireless Features**
Stargate	Intel PXA255 XScale Processor (400MHz)	32MB Flash 64MB RAM	802.11 Compact Flash 802.15.4 through MICA2 Interface
Imote2	32-bit PXA271 Marvell Processor (13-416)MHz	32MB Flash 64MB RAM	Integrated 802.15.4
MICA2/ MICAz	Atmel ATmega 128L 8-bit Micro-controller (8MHz)	128KB Flash 4KB RAM	2.4GHz, 802.15.4 Compliant Radio 250kbps Data Rate
TelosB	TI MSP430 Processor (8MHz)	48KB Flash 10KB RAM	2.4GHz, 802.15.4 Compliant Radio 250kbps Data Rate
PowWow	TI MSP430 Processor Actel IGLOO Low-Power FPGA	60KB Flash 2KB RAM	2.4GHz, 802.15.4 Compliant Radio 250kbps Data Rate
SHIMMER	TI MSP430 Processor (8MHz, 16 bit)	2GB Flash 2KB RAM	2.4GHz, 802.15.4 Bluetooth RN-42 250kbps Data Rate
WiseNet	Cool-RISC Processor (8MHz, 16 bit)	22KB RAM	Sub-Giga, 802.15.4 CMOS-Integrated Transceivers 25kbps Data Rate
Tyndall/UCC 25mm/10mm Node	Atmel ATmega 128/ 8051 8-bit Micro-controller	4KB RAM 4KB EEPROM	Sub-Giga, 802.15.4 Integrated Transceivers 50kbps Data Rate

Objective Modular Network Testbed in C++ (OMNeT ++) [123] is an object-oriented C++ based discrete event simulator. An OMNeT++ model consists of modules performing communication by passing messages. The active modules are known as simple modules and C++ is used to write them by utilizing the simulation class library. List of different simulation models available in OMNeT++ are TCP/IP networks, peer to peer networks, LAN/MAN protocols, Ad-hoc, wireless and sensor networks [71]. Couple of frameworks were developed under OMNeT++ such as Mobility Framework (MF) [3] and Power Aware Wireless Sensor Networks (PAWiS) [5]. The Mobility Framework supports simulation of wireless and mobile networks within OMNeT++ and it implements the supports for node mobility, dynamic connection management and wireless channel model. On the other hand, the PAWiS simulation framework facilitates the simulation of wireless sensor networks as it includes the internal structure of the sensor nodes as well as the network connecting them. The WSN node in PAWiS is splitted into functional blocks like a physical, MAC, routing, application layers, CPU, timer, AD/DA converters etc. Each of these blocks is mapped to module in simulation. The work flow is such that

at the beginning the node is composed of one module from the module implementation library. Then the modules are configured and in the next step the model is simulated and the results are evaluated. Based on the results the individual models could be refined. Out of these frameworks, PAWiS is a suitable option for wireless sensor networks simulation as most of WSN applications require limited mobility.

NS-2 [4] is a discrete event driven network simulator and is used in the simulation of the routing and multi-cast protocols, it supports an array of popular network protocols. One of the advantage of *NS-2* is the larger community that is working on the development of new and improving the existing features that are required in WSN.

Castalia [88] is a wireless sensor networks simulator for early-phase algorithm/protocol testing built at the Network and Pervasive Computing Program of National ICT Australia. It supports channel and radio model and provides support for defining physical process. It also supports the enhanced modeling of the sensing devices and other often-neglected attributes of WSN such as the node clock drift. In Castalia, the channel/radio model is based on empirically measured data, multiple transmission power levels are available and it has physical process models.

TOSSIM [78] is a TinyOS-based simulator for wireless sensor networks and it comes with standard installation of the operating system. It uses the *nesC* language that is a dialect of C used for the motes. TOSSIM simulates TinyOS network stack at the bit level, allowing experimentation with low-level protocols and top-level application systems. It supports network of hundreds or thousands of motes. The same code that is written for TOSSIM can be directly implemented onto the motes. Scalability is one of the issue with TOSSIM as it can only simulate code for TinyOS. It only supports mote-like devices implemented for TinyOS but not other more powerful nodes (e.g. A stargate). Further, it cannot simulate different binaries running on different motes.

SENSE [6] is a wireless sensor simulator that can be used by high-level users, network builders and the component designers. It uses a component port model which frees the simulation model from inter-dependence.

These simulation frameworks look to be an interesting choice but lack of technical support is the main hurdle to utilize them for wireless sensor networks research.

WSNet [27] is another event-based node simulator for wireless networks, which is used for node and environment simulation and it is developed by the INRIA research center in Lyon, France. In WSNet, the simulated nodes are built as an arbitrary assembly of blocks which represent either a hardware component, a software component or a behavior/resource of the node. There is no restriction in the number of blocks or the relation between the blocks. The blocks may model the following components/ behaviors:

- Mobility
- Energy source
- Application
- Routing protocols
- MAC protocols
- Physical Layer
- Radio interface
- Antenna

Each block is formally called as *bundle*, whereas each component is called as *entity*. WSNet does not have big community support but it is based on a simple modular approach and fundamental modules are available at each entity which are useful for comparing the algorithm or protocol. The construction of a new module is also simple, there are some modular functions which provide interface between the layers as well as for the formation of a node (more details can be found in [27]).

One of the key advantage of using WSNet simulator is that there is an associated node platform simulator WSIM [26], which allows to simulate different components of the sensor node. It relies on cycle accurate full platform simulation using microprocessor instruction driven timings. The simulator is able to perform a full simulation of hardware events that occur in the platform and to give back to the developer a precise timing analysis of the simulated software. Fig. 2.10 shows the integrated environment of WSNet and WSIM. In this thesis WSNet/WSIM network simulators are used for the development of a new MAC protocol.

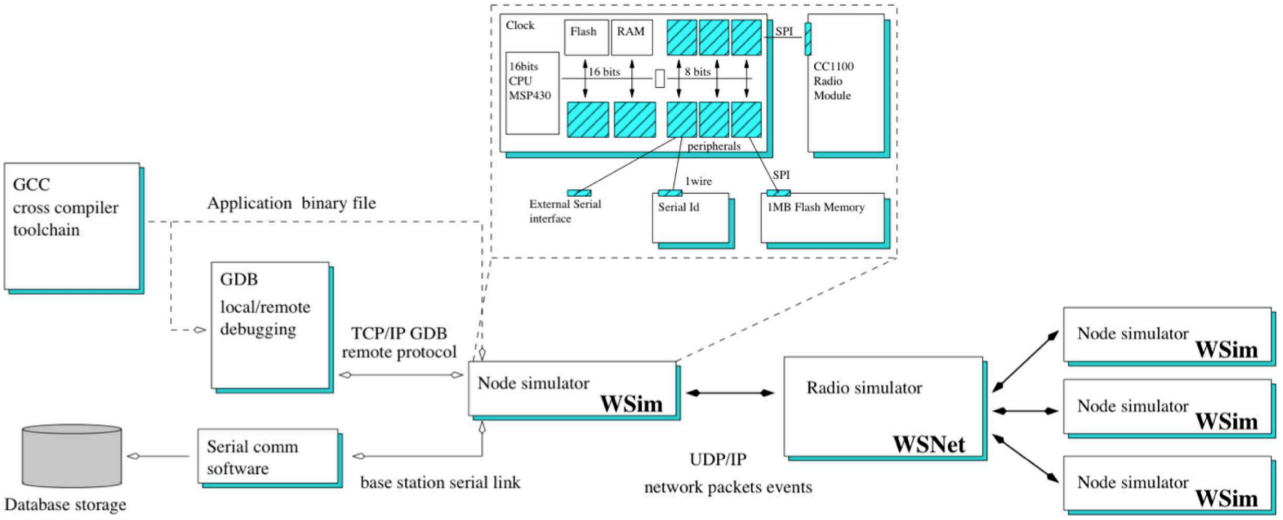


FIGURE 2.10: WSNet/WSIM simulator integrated design flow extracted from [27].

2.5 Conclusion

In this chapter an overview of different important aspects of WSN design were presented. First, a brief and focused survey of preamble sampling MAC protocols (such as BMAC, XMAC WiseMAC) is explained. Second, a state-of-the-art on various power models is presented and finally, several existing network simulators are explored.

In this thesis we have focused our attentions towards an accurate energy modeling and further on optimization techniques that can help to significantly reduce the energy consumption and increase the lifetime of the sensor networks. In this regard, special attention has been given on the dynamic aspects such as traffic and signal to noise ratio variations in the designing of a new MAC protocol and dynamic power optimization respectively. A hardware/software platform PowWow [63] is used to realize accurate power measurements, whereas, WSNet simulator is extensively used for the design of new MAC protocol and its verifications. The first step towards these optimizations is a reliable and accurate energy consumption modeling for WSN, that is introduced in the next chapter.

Chapter 3

Accurate Energy Consumption Modeling of WSNs

Wireless sensor networks (WSNs) are characterized by collaborative information transmissions from multiple sensor nodes observing a physical phenomenon [12]. Battery-driven sensor nodes are easy to deploy but they are severely constrained on physical size, cost and energy efficiency. Since the most important constraint in WSN remains the energy consumption, it is of great concern to know how to increase the life time of the sensor nodes. In this regard, it is pre-requisite to evaluate precisely the energy consumption of the sensor nodes in their network environment. Since the energy consumed is not only due to the components at the individual layers [107], different cross-layer energy models such as [131], [124], and [34], presented partial interaction between lower layers such as DLL (Data Link Layer), MAC (Medium Access Control) and PHY (Physical Layer). In this thesis we also considered lower layer interactions based on the parameters that influence the energy consumption.

We consider the global energy consumption as a function of MAC, DLL, PHY, and HWP (hardware platform parameters). The driving parameters of these layers in terms of energy consumption are the distance between the nodes, error correcting codes used, idle listening time, amount of collisions, retransmissions, overhearing, overheads, and the associated power consumption of the hardware components is the main focus of the global energy evaluation.

Energy consumption is critical to the occurrence of the events and more precisely to their timings. For example, in distributed asynchronous sensor networks (DASN), a sensor node can wake-up at a particular instant and communicate with another node without any collision, but the same node can have collision when it wakes up at another time instant. Consequently, the energy consumption for the same node is completely different. Therefore, to have deeper and more a realistic evaluation of energy consumption, it is important to identify different executing scenarios.

In this chapter, we propose a scenario-based hybrid and precise energy model with the aim to optimize the global energy of WSNs. The energy model relies on the interaction of both analytical modeling and real-time measurements. Firstly, the analytical model provides a global view of various elements of the DLL and MAC layers and shows their impact on the energy consumption. Secondly, the real-time measurements provide an accurate estimate of the power consumption of the software as well as the hardware components of the PowWow platform [63].

The model explores the detailed insight of the energy consumption based on different scenarios that can occur during the communication between sensor nodes. Each scenario is executed with regard to particular instantaneous events and we calculate the energy consumed through these scenarios, which helps to evaluate precise and detailed energy consumption of the software and hardware. Moreover, the model can be utilized off-line by knowing the statistics of the traffic as it will be done later in this thesis. It is important to mention here that the presented model focuses on a specific MAC protocol of the preamble sampling category but it can be easily extended in a larger context to other MAC protocols of this category.

The rest of this chapter is organized as follows. Immediately after this introduction a state of the art on energy consumption modeling is presented, followed by analytical models of the low level layers such as data link, MAC and physical layers. After that, the protocol stack implementation and the power measurements of PowWow platform are explained. Further, a scenario-based hybrid energy consumption model that combines the analytical models as well as the real measurements is presented. The model is validated and some of the applications of the proposed energy model are presented. This chapter ends with a brief conclusion and some improvements along with a motivation of further work which is presented in next chapters.

3.1 Related Work

Understanding power consumption is crucial for the design of autonomous nodes with the highest lifetime. During the last years, different power models have been presented and most of them are either only state based with no or limited interaction of cross layers, or based on hardware components only. Work such as [104], [96], and [69], presents energy model based on the consumption at the transmit and receive states and are limited to radio consumption. [99] presents an energy model which only takes the consumption of the hardware components into account, whereas [40] presents also a software energy model. Moreover, couple of companies, e.g. [30] and [31], have presented similar power consumption models. State-based models consider constant current levels in the different states, which does not represent a precise measure, as they lack the inside details of the variations in the current levels.

Now, with regard to the global energy model, [82] presents the MAC And Physical LAyer Power ($MAPLAP$) model (which is explained in Sec. 2.3.2, however, here we will present a critical analysis of MAPLAP model and more generally state-based power models) taking some cross layer impacts into account, but which is limited and again based on the different states. Some other works propose new methods [76] [109] to give accurate consumption models based on microprocessor platforms running on the widely used tinyOS event-driven operating system. These tools capture the detailed energy requirements of the CPU, radio, and every other peripherals, but the model presents very general overview and lacks real-time measurements. The authors in [131] propose a very interesting framework for an integrated layer design of the link and MAC layers of a sensor network in order to consider a global system for energy optimization of different layers. This is a solution to the fact that some components are not selected/designed knowing the impact of cross-layer interactions, especially concerning the power consumption evaluation.

As far as accurate power estimation methods and techniques are concerned, several models, such as [21], [67], and [82], represent the prediction of the node current consumption in terms of numerical equations with a large number of parameters. These parameters evaluation is not always trivial, also the accuracy of the predictions of these static models lacks reliability. Similarly, node platform vendors provide tools for node lifetime calculation [36] based on the consumption of the charge consumed as a function of node duty

cycle, which is defined as the ratio of active and non-active node time intervals [22]. The main shortcoming with such an approach is that it assumes a deterministic current usage profile to predict node life time. An alternate solution can be a monitoring tool to characterize the node current usage profile as presented in [64]. For the precise power consumption of the WSN hardware platform, [97] presents a very interesting measuring methodology as well as the power consumption for elementary functionalities (such as microcontroller, internal flash, LEDs, timer and radio components) of the WSN platform.

The power and energy models in [104], [99], and [69] are based on limited aspects and propose very general overview. The work proposed in [131] is in fact a framework for partial cross-layer integrated energy model, but it does not provide a real measure of the energy. So, the bottom line is that, there exists a deficiency for realistic energy model which could provide the details of the energy consumption and explain all the possible scenarios that can occur during the communication in sensor networks, which eventually leads us to the inside details of the state-based models.

3.2 Partial Cross-Layer Modeling

In this section the analytical characteristics of different low level layers with special focus on the MAC layer is presented. Different parameters such as *wake-up interval* of a node, *collision probability* and *amount of data transmitted* from node i to node j during a given time interval, have a direct influence on the energy model that is presented later in Sec. 3.4.2. For the context of this part of the thesis, the parameters considered fixed are the topology and geographic configuration, while the other parameters (like data error rate, wake-up intervals, packet collisions, packet size, transmission power level, etc.) that have to be investigated are associated with the low-level layers such as DLL and MAC layers.

3.2.1 Data Link Layer Model

The link layer model is based on the error control strategy which can be applied mainly through three different strategies; automatic repeat request (ARQ), forward error correction (FEC) and hybrid automatic repeat request ($HARQ$). The objective of our link-layer model is to manage the automatic repeat request (ARQ) and the forward error coding (FEC) to ensure a reliable link. The quality criterion considered is the

tolerance to binary errors. Now, based on the bit error probability P_b [77] [110], the packet error rate (PER) for the ARQ and FEC schemes are presented below. The ARQ scheme is based on cyclic redundancy check (CRC) for the error detection. The PER for the payload pl (which is the actual data in the complete data packet) can be represented as

$$PER_{ARQ}(pl) = 1 - (1 - P_b)^{pl} \tag{3.1}$$

A detection-only strategy can not tolerate any binary error, while a correction strategy associated with a simple checksum for detection can tolerate one error. In the case of forward error correction coding, the link layer can be designed using several techniques such as block and convolutional codes. For the case of block codes, considering that the payload is pl and if the tolerance (i.e. the number of errors that can be corrected by the FEC) is noted as tol, then the re-transmission rate (RT) for the communication between node i and node j for a tolerance tol is calculated as

$$RT_{i,j}(tol) = 1 - \left(\sum_{k=0}^{tol} C_{pl}^{k} \cdot BER^{k} \cdot (1 - BER)^{pl-k} \right) \tag{3.2}$$

where $C_{pl}^{k} = \binom{pl}{k}$ is the binomial coefficient of pl and k.

3.2.2 Medium Access Control Model

Medium Access Control is the ability of a node to efficiently share the wireless medium with the other nodes in the network. In the context of global energy consumption, the MAC layer plays a pivotal role in optimizing the WSN lifetime, because the dominant energy consumption is due to the radio transceiver chip whose activity is controlled by the MAC layer [19]. Another fact is that the actual payload at the MAC layer is negligible with regards to the associated headers. As a consequence, in typical WSN applications such as temperature monitoring, surveillance, etc., the radio wakes up at a more frequent rate with regards to what it actually transmits, and that leads to a significant proportion of energy waste.

Preamble sampling MAC protocols are an energy efficient class of MAC protocols as explained in Chapter 2. In the preamble sampling protocols the wake-up time of one node is independent of the other nodes in the network, which means that two nodes can wake-up and send their preambles and subsequent data packets at the same time,

which results in a *preamble collision*. In addition to *preamble collisions*, *data collisions* can occur due to randomly distributed sensing and also due to hidden terminals, even though in different variants of preamble protocols, such as XMAC, WiseMAC, TICER, RICER etc., the data packet is being sent after channel sensing and initial rendez-vous between the nodes. So, in order to derive the mathematical expressions for the probability of collisions, we define some parameters: consider n_r being the number of nodes in the radio range, I_{wu} is the wake-up interval, $PTime$ is the duration of the preamble, and T_{wu} (wake-up time) is the time during which the receive node remains awake for the data as shown by the transmitter initiated communication in Fig. 2.2.

Let us consider that the network traffic is approximated as a Poisson distribution and it has a mean value of data generated per second per node as λ. Every node senses the channel before transmitting the preamble and data packets. Now, in the RICER protocol a node wakes up in the receive mode at every wake-up-interval and sends the wake-up-beacon (WUB). After that, the receive node waits for the duration of wake-up time to receive the response. Let denote T_{obs} as the observation time to observe the traffic and n_r the number of nodes which are in the radio range. Then, the total number of rendez-vous (n_{rdv}) during T_{obs} is

$$n_{rdv} = n_r \frac{T_{obs}}{T_{wu}}. \tag{3.3}$$

and the occupation rate (OR) of the channel is

$$OR = n \frac{T_{wu}}{I_{wu}}, \tag{3.4}$$

where n is the total number of nodes in the network. OR must be inferior to 1 in the general situation or equal to 1 when the channel is saturated. For a particular node, the probability of *preamble collision* (Pr_{PC}) is equal to

$$Pr_{PC} = (n_r - 1) \frac{T_{wu}}{I_{wu}}. \tag{3.5}$$

After the general overview of the preamble sampling category of MAC protocols, we will focus in this work on cycled receiver protocols for a detailed MAC energy analysis. The chosen protocol is based on the principle of Receiver Initiated CyclEd Receivers (RICER) [81], [80]. RICER is a pseudo-asynchronous technique to realize rendez-vous between wireless nodes. It means that nodes establish rendez-vous on demand, but

underlying there exists a periodic wake-up scheme. RICER MAC scheme can be used with very simple communication hardware with the following key features:

- Reduced preamble size and signals, *beacon*, *data*, and *acknowledgment* frames.
- Low-power pseudo-asynchronous mechanism ideally suited for low traffic, i.e. WSN.
- Only the beacon frame is broadcasted, therefore it results in limited overhearing.
- Reduced idle monitoring and collision with underlying self periodicity.
- RICER performance is better than its counter-part, Transmitter Initiated CyclED Receiver (TICER) under strong fading conditions [82].

Several variants of RICER and their comparison with Transmitter Initiated CyclED Receiver (TICER) have been presented in [82]. We have chosen RICER3 (3-way handshake), which is shown in Fig. 3.1, because it contains less preambles (control signals) and its performance is comparable with other variants. It is important to mention here that, RICER3 is a power optimized protocol for low traffic WSN application [82], but it can be also very effective for adaptive traffic load providing that the wake-up interval is adjusted adaptively.

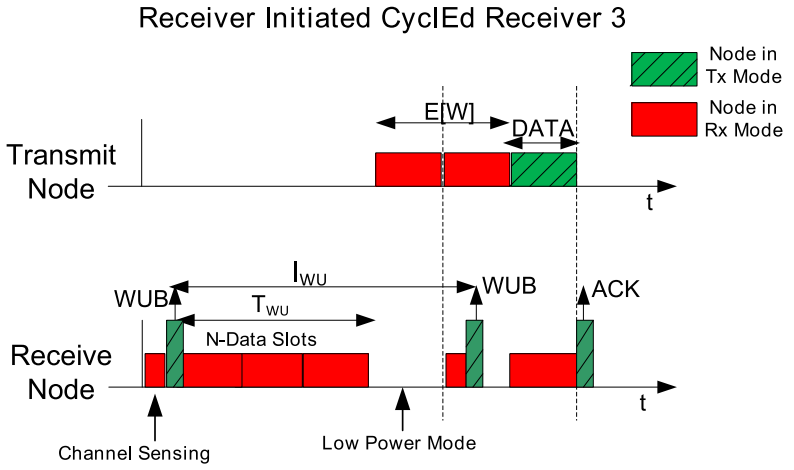


FIGURE 3.1: RICER3 is a 3-way handshake receiver initiated cycled receiver scheme. The N-data slots are placed in order to avoid the collision which happens when two nodes send the data to the same destination node at the same time in response to the wake-up-beacon. $\mathbb{E}[w]$ is the expected waiting time, T_{wu} is the total wake-up time of the receiver, I_{wu} is the wake-up interval of each node, WUB is the wake-up-beacon signal and Ack is the acknowledgment signal.

3.2.2.1 Collision Probability Model

In order to evaluate the collision probability for this pseudo-asynchronous protocol, the MAC layer has to be accurately described. In RICER3, as shown in Fig. 3.1, N-data slots are placed in order to avoid the collision which happens when two nodes send the data to the same destination node at the same time in response to the wake-up-beacon. The destination node wakes up with its own periodic interval and senses the channel for a very short time before broadcasting the wake-up-beacon, in order to avoid the control packet collisions on the channel. Whereas the source node on reception of a wake-up-beacon also senses the channel for a short duration before transmitting the data to prevent data collision. Data collision can occur when two or more source nodes select the same data slot of the destination node. It is being observed that the channel sensing time needs to be at least as long as twice the packet propagation time [80]. Now, in order to reduce data collision, it is necessary to choose an appropriate number of data slots in RICER3. The more slots there are, the lower the data collision rate and thus the fewer re-transmissions, but the more monitoring power wasted per wake-up (in the case of limited data transmission). These factors lead to a trade-off in both power consumption and latency. It is observed that on a low traffic wireless sensor network application, the probability of three nodes to transmit at one node at the same time is very rare [80] [63]. So we have assumed in our collision model that there can be maximum two nodes that can transmit the data to the same destination node at the same time.

Two kinds of collision are possible in this pseudo-asynchronous scheme:

- The control packet collision, called wake-up collision (WUC), can happen at the initial phase of rendez-vous when the two nodes wake-up in the receive mode at the same time, since both nodes found the channel as idle and therefore, send the wake-up-beacon at the same time.
- The data packet collision, called data collision (DC), is possible when the two source nodes are intending to transmit the data to the same destination node. As they receive the wake-up-beacon from the destination node, both source nodes find the channel being idle and transmit their data in the same data slot.

Wake-Up Collision In the RICER protocol, a node wakes up in the receive mode at an interval called wake-up interval (I_{wu}) and sends the wake-up-beacon (WUB). After that, the receive node waits for the duration of wake-up time (T_{wu}) to receive the response. Then, the probability of the wake-up collision is equal to the sum of collisions due to $n_r - 1$ nodes plus the one due to a hidden node. Then the expression can be calculated as

$$Pr_{WUC} = (n_r - 1)\frac{T_{wu}}{I_{wu}} + \frac{T_{wu}}{I_{wu}} = n_r\frac{T_{wu}}{I_{wu}}. \tag{3.6}$$

Data Collision The probability that i source nodes (out of n_r nodes which are within the radio range) intend to transmit to the same destination node at the same time is

$$Pr(i) = \frac{(n_r \cdot \lambda \cdot I_{wu})^i}{i!}\exp(-n \cdot \lambda \cdot I_{wu}). \tag{3.7}$$

Focusing on one of these i source nodes, with N data slots, the probability that the data of the node does not collide with any others is

$$Pr_{success} = \frac{N(N-1)^{i-1}}{N^i} = \left(\frac{N-1}{N}\right)^{i-1}. \tag{3.8}$$

Accordingly, the total probability of data collision Pr_{DC} for this particular node in T_{obs} is

$$Pr_{DC} = \sum_{i=2}^{n_r} Pr(i)\left(1 - \left(\frac{N-1}{N}\right)^{i-1}\right). \tag{3.9}$$

Note that, Pr_{DC} is expected to be higher than the actual *data collision rate*, because nodes sense the channel for a random time before transmitting a data. On the other hand, it does consider the collision due to the hidden terminal. So in this way, as long as the attempting source nodes are in the radio range, source nodes that sense later will detect the channel as busy and can back off.

3.2.2.2 Analytical Analysis of Power Consumption in RICER3

Let N be the number of data slots followed by each wake-up-beacon (WUB), then the power consumption for RICER3 can be calculated as [80]

$$\begin{aligned} P_{tot} = \Delta_{tx}P_{tx} + \Delta_{rr}P_{rr} + \Delta_{mn}P_{mn} + \Delta_{aq}P_{aq} + \Delta_{sp}P_{sp} \\ + \sum \mathbb{E}[N_{turn}]P_{turn} \end{aligned} \tag{3.10}$$

where $\Delta_{tx}, \Delta_{rx}, \Delta_{aq}, \Delta_{mn}$, and Δ_{sp} are the percentage of the time each node spends in the transmit, receive, acquisition, monitoring, and sleep states respectively. These are

the five different states represented in the MAPLAP model [81]. The term $\mathbb{E}[N_{turn}]$ is the expectation of the transition from one state to another state of a node within one second. $P_{tx}, P_{rx}, P_{aq}, P_{mn}, P_{sp}$, and P_{turn} are the power consumption levels in the transmit, receive, acquisition, monitoring, sleep, and transition states respectively and are defined by:

$$¶_{tx} = \mathbb{E}[N_{WB}]I_{wu} + \lambda^{'}(T_{DATA} + T_{ACK}) \tag{3.11}$$

$$¶_{rx} = \lambda^{'}(I_{wu} + T_{DATA} + T_{ACK} - 3T_a) \tag{3.12}$$

$$¶_{mn} = \mathbb{E}[N_{WB}]NT_{DATA} + \lambda^{'}\mathbb{E}[W] \tag{3.13}$$

$$¶_{aq} = \lambda^{'}(3T_a) \tag{3.14}$$

$$¶_{sp} = 1 - ¶_{tx} - ¶_{rx} - ¶_{aq} - ¶_{mn} \tag{3.15}$$

where $\mathbb{E}[N_{WB}]$ is the expected number of WUB transmitted during T_{obs} and is equal to

$$\mathbb{E}[N_{WB}] = (1 - Pr_{busy})\frac{(1 - \lambda^{'}(\mathbb{E}[W] + 2T_{DATA} + 2T_{ACK})}{T_{obs}}.$$

T_{DATA} and T_{ACK} are the duration of data and acknowledgment transmission. Pr_{busy} is the probability that a node detects the channel to be busy before transmitting the WUB, $DATA$, or ACK frame and is expressed as

$$Pr_{busy} = 1 - (1 - (\mathbb{E}[N_{WB}]I_{wu} + \lambda(T_{DATA} + T_{ACK})))^{n_r}.$$

In the case of the channel being busy, the node backs off for one or multiples of I_{wu}. The channel is detected to be busy if any other packets are already transmitted on the network. After taking account of collisions, the actual traffic load becomes $\lambda^{'}$ and can be calculated as

$$\lambda^{'} = \frac{\lambda}{1 - Pc}$$

where Pc includes both Pr_{WUC} and Pr_{DC}.

$\mathbb{E}[W]$ is the expected monitoring time (or expected waiting time) of the source node until it receives the WUB from the destination node. This monitoring time can increase in two cases. First, if the destination node finds the channel being busy before transmitting its WUB, in this case the transmitting node has to wait for multiples of I_{wu} (but the transmitter can send its own WUB to avoid the longer latency for the whole network). Second case is due to the collision of packets over the channel.

The power consumed during the switch between the states depends upon the expected number of transitions from one state to another. These can be:

- *Sleep-to-transmit* ($\mathbb{E}[N_{SP-TX}]$)
- *sleep-to-receive* ($\mathbb{E}[N_{SP-RX}]$)
- *transmit-to-receive* ($\mathbb{E}[N_{TX-RX}]$)
- *receive-to-transmit* ($\mathbb{E}[N_{RX-TX}]$)

During one second for RICER3 the expected number of transitions can be calculated as [80]

$$\mathbb{E}[N_{SP-TX}] = 0 \tag{3.16}$$

$$\mathbb{E}[N_{SP-RX}] = \frac{(1-\lambda^{'}(\mathbb{E}[W]+2T_{DATA}+2T_{ACK}))}{I_{wu}} + \lambda(1+1) \tag{3.17}$$

$$\mathbb{E}[N_{TX-RX}] = \frac{(1-\lambda^{'}(\mathbb{E}[W]+2T_{DATA}+2T_{ACK}))}{I_{wu}} + \lambda \tag{3.18}$$

$$\mathbb{E}[N_{RX-TX}] = \frac{(1-\lambda^{'}(\mathbb{E}[W]+2T_{DATA}+2T_{ACK}))}{I_{wu}} + \lambda(1+1) \tag{3.19}$$

3.2.3 Physical-Layer Model

Though the IEEE.802.15.4 standard is modeled with offset quadrature phase shift keying ($OQPSK$), we have chosen a generic approach for the modulation scheme i.e. M-ary phase shift keying (PSK). The M-ary PSK modulation schemes have been used for analytical analysis and the expressions are given for the error probability in terms of *bit* (P_b) and *symbol* (P_s). The digital phase modulated signal waveform can be represented as

$$s_m(t) = g(t)\cos\left[(2\pi f_c t + \frac{2\pi}{M}(m-1))\right] \tag{3.20}$$

where $1 \leq m \leq M$, and $0 \leq t \leq T$.

The symbol error rate P_s of M-ary PSK signals for additive white Gaussian noise ($AWGN$) channel is expressed as [110]

$$P_s = 1/\pi \int_0^{(M-1)\pi/M} \exp\left(-\frac{kE_b}{N_o}\frac{\sin^2[\pi/M]}{\sin^2(\theta)}\right) d\theta \tag{3.21}$$

where

$$\frac{Eb}{No} = SNR\frac{W}{R}$$

with SNR the signal to noise ratio, R the bit rate and W the bandwidth. Now, from [77] and [110], the bit error probability P_b can be approximated as

$$P_b = 1/k\left(\sum_{i=1}^{M/2}(W_i^{'})P_i\right) \tag{3.22}$$

where $W_i^{'} = W_i + W_{M-i}$, $W_{M/2}^{'} = W_{M/2}$, with W_i the Hamming weight of the bits assigned to symbol i and $M = 2^k$. Now, P_i is equal to

$$P_i = \frac{1}{2\pi}\int_0^{(1-(2i-1))\pi/M}\exp\left(-\frac{kE_b}{N_o}\frac{\sin^2[(2i-1)\pi/M]}{\sin^2(\theta)}\right)d\theta - \frac{1}{2\pi}\int_0^{(1-(2i+1))\pi/M}\exp\left(-\frac{kE_b}{N_o}\frac{\sin^2[(2i+1)\pi/M]}{\sin^2(\theta)}\right)d\theta.$$

In the special case of $M = 2$, i.e. $BPSK$, the error probability is given by [111] as

$$P_s = P_b = \int_{\sqrt{2E_b/N_o}}^{\infty}\frac{1}{2\pi}\exp\left(-\frac{u^2}{2}\right) = Q\left(\sqrt{2E_b/N_o}\right). \tag{3.23}$$

Also for, $M = 4$, i.e. $QPSK$, the expression for the error probability of symbols and bits are given by [98]

$$P_s = 2Q\left(\sqrt{2E_b/N_o}\right)\left[1 - 1/2Q\left(\sqrt{2E_b/N_o}\right)\right] \tag{3.24}$$

$$P_b = Q\left(\sqrt{2E_b/N_o}\right) \tag{3.25}$$

It is noteworthy that the expression for probability of error for $BPSK$, $QPSK$ and $OQPSK$ are theoretically the same values. Though, as M increases further, the SNR has to increase to achieve the same probability of error [98].

We have chosen a generic approach to evaluate different modulation schemes based on analytical expressions for power efficiency. The modulated expressions for the $OQPSK$ and $QPSK$ are nearly the same, the only difference existing in the phase alignment of d_i and d_q which are the in-phase and quadrature components of the modulated signal as explained in [111]. Similarly, the probability of error for the $BPSK$, $OQPSK$, and $QPSK$ schemes gives an expression of the BER (binary error rate) that can be observed in the bit stream as

$$BER = 1/2.erfc(\sqrt{SNR})\cdot \tag{3.26}$$

3.2.4 Channel Model

The channel model used is an additive white Gaussian noise. Considering that the signal power before the transmission is P_{Level} (expressed in dB) and the gain of the transmitting antenna is G_t (dBi), then the radiated power i.e. P_{rad} (dB) is

$$P_{rad} = P_{Level} + G_t. \tag{3.27}$$

Now, at a distance of 1 meter from the antenna, the signal power can be expressed as

$$P_{1m} = P_{rad}.\lambda_c^2/16\pi^2, \tag{3.28}$$

with λ_c the carrier wavelength. For a distance $D_{i,j}$ meters between the transmitter and the receiver, the power loss P_d lowers the signal strength and is expressed as

$$P_d = P_{1m}/D_{i,j}^{\alpha}, \tag{3.29}$$

where α is the attenuation coefficient [57]. Further, at the receiver end, the attenuated received signal is boosted or attenuated by the gain of the receiver antenna and the expression for the received signal-to-noise ratio (SNR) can be deduced as

$$SNR = P_{rad} - P_d + G_r - P_n \tag{3.30}$$

with P_n is the noise power in dBm and G_r the gain of the receiver antenna expressed in dBi.

3.2.5 A Brief Conclusion on the Analytical Model

To summarize the analytical model, the expressions for different parameters of lower layers that have a significant impact on the global energy consumption in WSN have been presented. Data link layer supports the model for error detection and correction. This error occurs during the transmission due to non ideal channel conditions. For that reason, the bit error probability for the $OQPSK$ modulation scheme computed, and the amount of retransmissions for a given tolerance have been presented. The MAC layer is the core of our model, which is based on energy efficient preamble sampling category and we have explored the collision probability models along with the power consumption for RICER protocol. The analytical model will be used along with the power measurements to obtain the hybrid energy model which is presented in Sec. 3.4.2.

3.3 Protocol Implementation and Power Measurements

In this section, different optimizations that have been performed at each layer and the results of the power measurements based on our low power WSN hardware platform are explained. The architectural block diagram of our WSN platform (PowWow [63]) is shown in Fig. 3.2. PowWow is a hardware platform associated to a software architecture designed for a complete WSN solution. The hardware platform is, like many others, based on a low-power micro-controller and a radio transceiver. However, PowWow also includes new features which improve the energy efficiency with regards to state-of-the-art platforms through dynamic voltage and frequency scaling (DVFS) in the digital processing part (micro-controller) and in co-processing part using a low-power FPGA (Field Programmable Gate Array). The detailed description of the hardware architecture and its utilization can be found in [23].

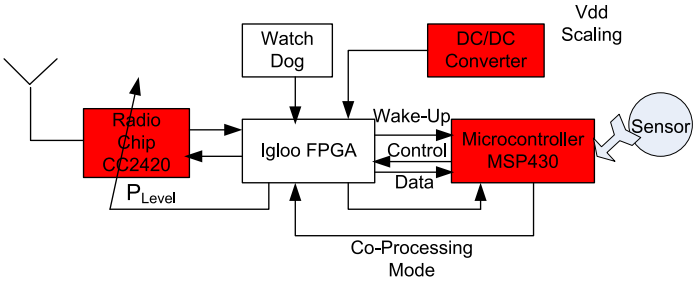


FIGURE 3.2: Hardware block diagram of PowWow. The key components in terms of power consumption are processing units (T.I MSP430, Actel IGLOO FPGA), DC/DC converter (used for voltage scaling), and the radio chip (T.I. cc2420).

The PowWow software is based on Contiki, which is built around an event-driven kernel but provides optional preemptive multi-threading that can be applied to individual processes [39], and more precisely on the protothread library [41], which enables the use of extremely light-weight, stack-less threads. Protothreads allow to realize event-driven systems, and it has been shown that asynchronous processing that is typical of sensor networks applications perfectly suits the event-driven programming paradigm.

Another famous component based event-driven operating system is TinyOS, which is widely used in the sensor networks community. One of the problems of TinyOS is that, it does not allow block/wait statements inside the executing event handlers. In fact the programs have to turn into an explicit state machine, also it is hard to program, understand and debug. On the other hand, with protothread we can write block/waits, inside the event handler and can also utilize the threads if needed. So, the Protothread

library has been chosen instead because of its greater flexibility and ease of development in C language.

The rest of this section presents some optimizations in the RICER protocol and then accurate power measurements using PowWow platform

3.3.1 Optimizations of MAC Parameters

The MAC protocol (RICER3) has been optimized in terms of packet size, collision reduction (wake-up-collision and data) and effective utilization of dual channels in comparison to [82]. The *overhead* in the data and control packets is one of the main source of energy utilization. Therefore, overheads have been reduced significantly in the actual implementation, the size of control packets, i.e. wake-up-beacon and acknowledgment is reduced to four bytes, whereas the data packet is consisting of sixteen bytes, as shown in Fig. 3.3.

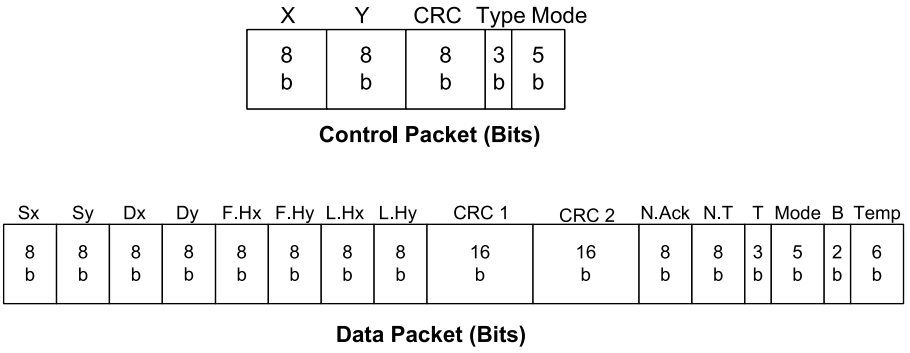


FIGURE 3.3: The Control Packet (wake-up-beacon and acknowledgment) consists of geographical network address (x,y) of the receive node which transmits the control packets, cyclic redundancy check (CRC) and the last byte contains the type of packet (3-bits) and the mode of communication (5-bits). The data packet consists of the network address of source and destination as well as the hop-addresses of the next and last hop-node, along with few bytes for acknowledgment number, temperature, battery, and $N.T$ is frame buffer indicator.

Two channels have been used quite effectively, *channel* – 1, works for the normal mode which supports low traffic such as temperature sensing, and *channel* – 2 is used when there exists a heavy data traffic such as images to be transmitted in typical surveillance monitoring or software updates.

Collision avoidance and *idle monitoring* are the primary concerns for power reduction in pseudeo asynchronous rendez-vous schemes. In RICER3, there is significant power

consumption due to wake-up-collision, as many WUB are transmitted to have successful rendez-vous. Many nodes can wake-up at the same time and can result in collision mainly because of clock drifts and asynchronous nature of the protocol. To reduce this wake-up-collision and to reduce the bad situation where two or more nodes wake-up always at the same time, a *random delay* has been introduced to each node. This *random delay* brings the small offset which is enough to reduce WUB collision significantly. Finally, *idle monitoring* has been reduced significantly by several ways. The data packet is always ready at the transmitter as it received the WUB and due to the fact that the time to switch between the states is very short, therefore, the receiver does not need to wait for the N data slots as shown in Fig. 3.1. In our implementation, the receiver only waits for the *first byte* of the data packet. This reduces the unnecessary wait/monitoring of the receive node for N data slots as in [81] [82], and [80]. At the same time, the probability of collision in the absence of N data slots will remain the same as (the) data collision is independent of data slots and it only occurs when two nodes transmit exactly at the same time. The optimized RICER3 is shown in Fig. 3.4.

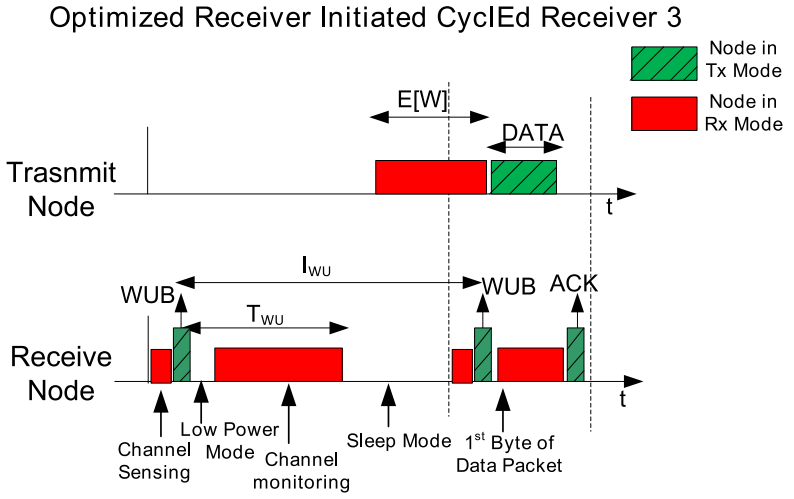


FIGURE 3.4: Optimized 3-way handshake receiver initiated cycled receiver scheme. The N-Data slots are not necessary to avoid the collision, instead the receiver only requires the first byte of the data packet. Therefore, the idle channel monitoring is irrespective of data slots but adjustable according to the application.

3.3.2 Platform Power Measurements

PowWow hardware platform is designed with modularity as its main characteristics. It is composed of a central printed circuit board (PCB) and of different daughter-cards as mentioned in [63]. The real-time measurements are conducted with the *Agilent* $N6705A$

DC Power Analyzer which is equipped with four channels modules which means that at one time the power consumption of four components/devices can be measured[1]. The platform setup for measurements is constituted of three WSN PowWow motes connected to the DC Power Analyzer as shown in Fig. 3.5. The external power supply of 3.3 *volts* is used. Four modules of the $N6705A$ are connected with the $cc2420$ and $MSP430$, to measure the current consumption of each component of the (connected) nodes. The three connected nodes are: a node in receive mode (*Rx. Node*) and two nodes in transmitter mode (*Tx. Node 1* and *Tx. Node 2*). It is noteworthy that no DC/DC conversion is required with this configuration which reduces the perturbation of the measurements. The parameters that have been described in Sec. 3.2, such as *collision avoidance*, *wake-up* and *data collisions*, have been identified through real-time measurements. Some general parameters used in the physical platform are shown in Tab. 3.1. These parameters are data length, MAC layer timing data, antenna gains, receive and transmit power levels of the radio transceiver chip.

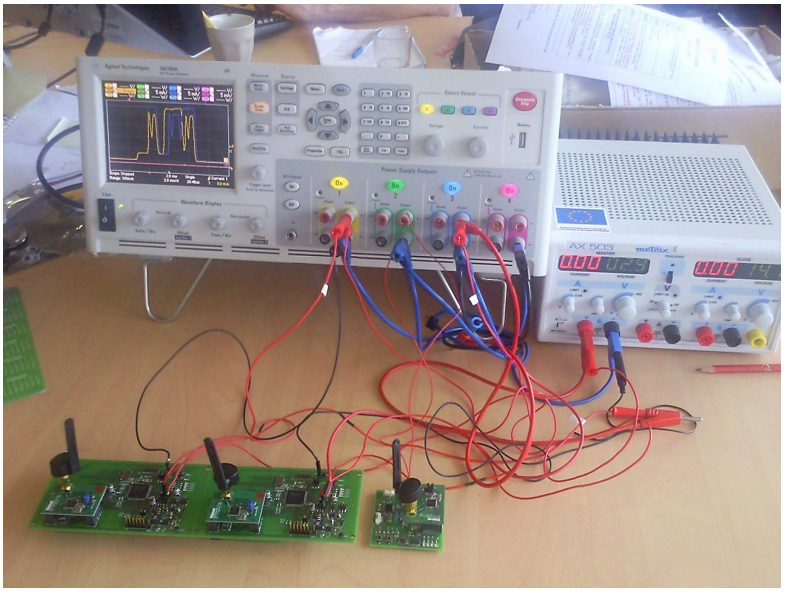


FIGURE 3.5: The platform setup for measurements is constituted of three WSN PowWow motes connected to the DC Power Analyzer. The external power supply of 3.3 *volts* is used. Four modules of the $N6705A$ are connected with the $cc2420$ and $MSP430$, to measure the current consumption of each component of the connected nodes as shown in Fig. 3.10.

The current consumption curves for the complete communication between three sensor nodes are shown in Fig. 3.6. In the MAC protocol, since the transmission is initiated by

[1]The power analyzer has only four channels for measuring the current consumption so at one time maximum 4 nodes can be connected. Therefore, it is very difficult to observe the real-time collisions of control and data packets, but after many attempts we have achieved some collisions and their results are presented later in Fig. 3.11.

pl = 128 bits	$\mathbb{E}[w](Tx)$ = (1.5- 15) ms
G_t = -1.0 dBi	T_{react} = 0.1 ms
G_r = -1.0 dBi	T_{sync}= = 0.1 ms
P_n = -95 dBm	I_{wu} = (0.3-3) s
F_c = 2.4 GHz	$T_{DF}(Rx)$ = (1.5-4.5) ms
C = 3.10^8m/s	$T_{DF}(Tx)$ = 1ms
α = 4	$P_t(0dBm)$ = 57.2 mW
P_r = 56.9 mW	$P_t(-20dBm)$ = 30.5 mW

TABLE 3.1: System parameters of the physical platform and the timing values of the MAC Protocol. T_{react} is the reaction time for switch the states, T_{sync} is the time to synchronize, $T_{DF}(Rx)$ is the time to receive the data frame, $T_{DF}(Tx)$ is the time to transmit the data frame.

the receiver, the transmitting node after waking up has to wait for the WUB from the receiver, before it starts transmitting the data. In our implementation, the maximum expected waiting time of the transmitter is 15 *blocks*, where each *block* can be between $0.3s$ to $3s$ depending on the application.

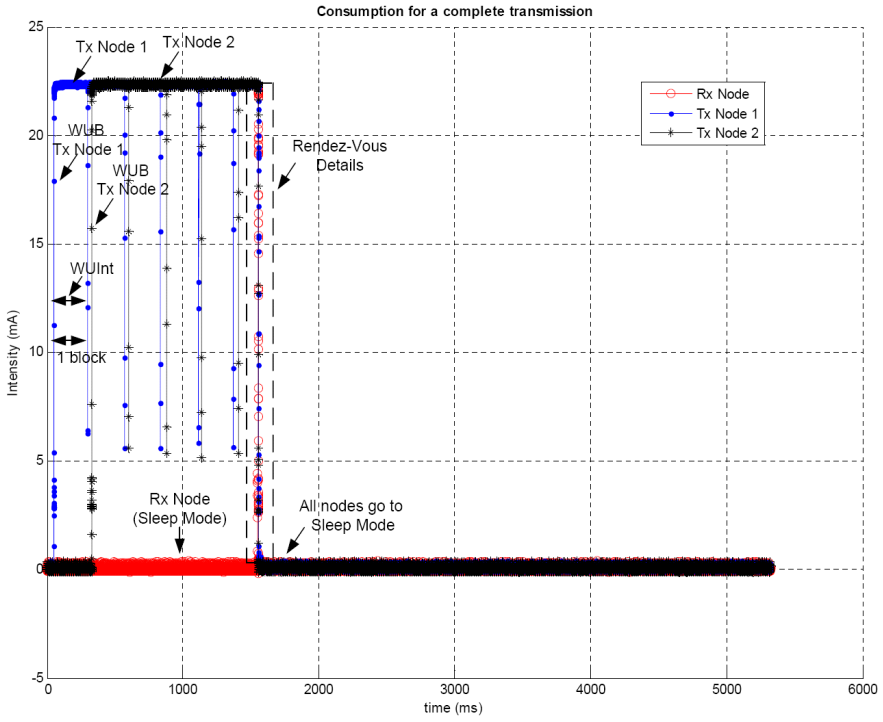


FIGURE 3.6: Total current consumption: two nodes want to transmit the data to the same destination node. In this particular case, after 5 *blocks*, the transmitter wakes up and sends its WUB, and a rendez-vous (detailed in Fig. 3.8) occurs between the two nodes on the single channel. The total current consumption including the $cc2420$ and $MSP430$ is $22.5mA$.

In Fig. 3.6 or 3.7, the combined total current consumption based on software and hardware ($MSP430$ and $cc2420$) for three WSNs nodes are shown. In this configuration each node is connected to a single power analyzer module. The interesting aspect of the

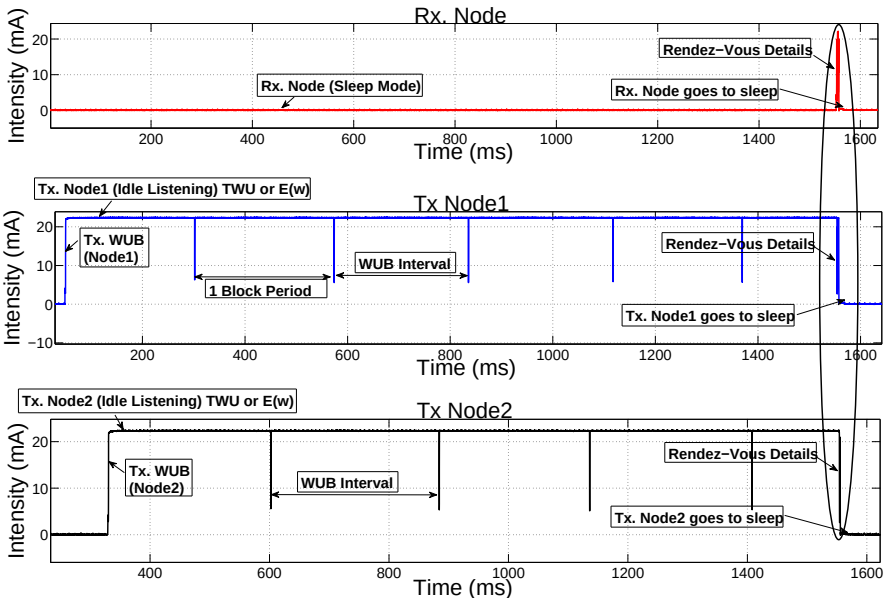


FIGURE 3.7: Same as Fig. 3.6, only three nodes are presented separately for clarity.

Fig. 3.6 (coming from MAC protocol) is that the transmitting nodes also send their own WUB at an interval of every *block*, so as to reduce the overall network latency and to improve the power consumption of idle monitoring of other neighboring nodes that want to communicate with the transmitting node. The rendez-vous occurs between the nodes and after the complete communication all the nodes go to sleep mode.

The details of a rendez-vous are shown in Fig. 3.8 or 3.9. Two nodes want to transmit data to the same destination node and both are waiting for WUB in the receive mode. The receive node wakes up (according to its own wake-up interval) from the sleep mode and configures its software as well as hardware in calculation before transmission (CBT). The execution of this calculation can be different depending upon the exact mode of software and hardware, for example, if the regulator of the radio transceiver is off (sleep state), it requires complete restart. Similarly if the micro-controller (software) is in deep sleep mode it requires to boot from the start. When the WUB is received, the two nodes change their mode from receive to transmit and start transmit their data packet. Since both nodes sense the channel through clear channel assessment (CCA) before transmitting their data, in this particular example, the *Tx. Node 2* finds the channel already occupied by *Tx. Node 1* and hence it backs off and therefore the data collision is being avoided through channel sensing as can be seen in Fig. 3.8. Meanwhile, the *Tx. Node 1* continues its transmission which completes at the reception of ACK signal from the receive node.

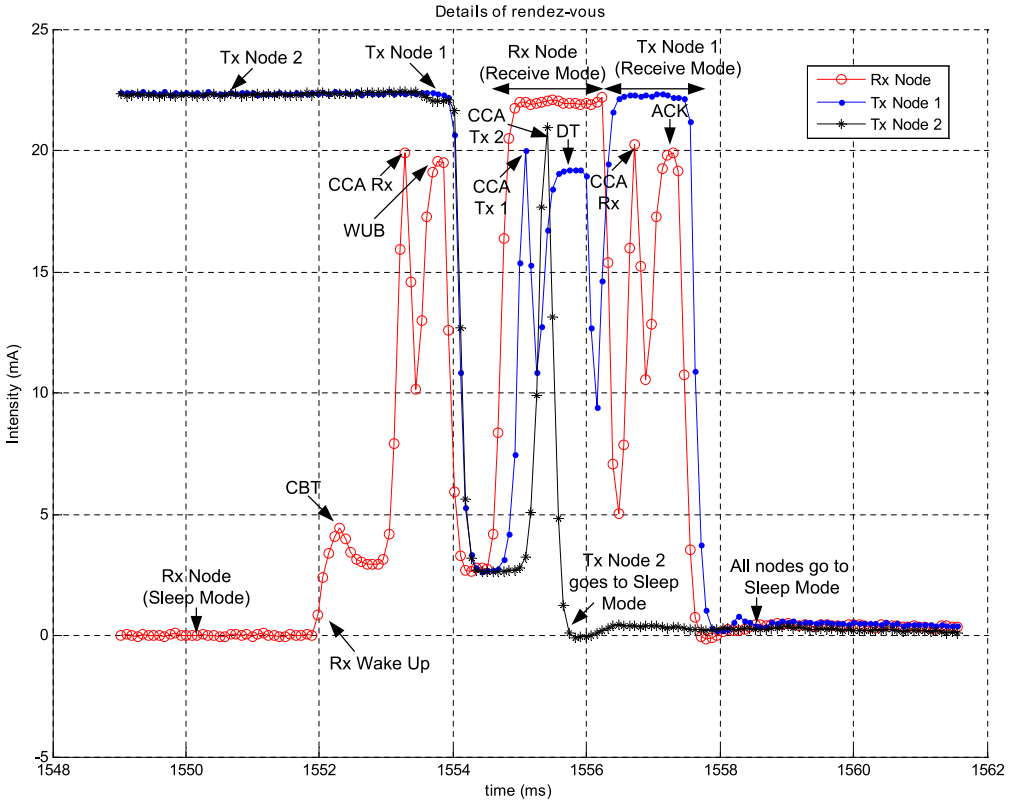


FIGURE 3.8: Details of a rendez-vous: three phases i.e. WUB, DT and ACK are clearly shown, and all the events are described during the rendez-vous. The Tx *node* 1 senses the channel first and hence occupies the channel before the Tx *node* 2. Therefore the collision is being avoided.

Though Fig. 3.6 and Fig. 3.8 show the total current consumption and provide the detail inside of the communication phases during rendez-vous, the details of sleep, low power, active, transitions, states are not clearly identified. For that matter, Fig. 3.10 shows the details of software as well as hardware consumption. The software component ($MSP430$) consumes 0.0 μA, 0.01 μA, 2.5 mA, and 3.7 mA, in sleep, low power mode (LPM5), transmit, and receive modes respectively. On th other hand, the hardware transceiver component ($cc2420$) consumes 0.0 μA, 0.4 mA, 17.6 mA, and 19.6 mA, in sleep, LPM, transmit, and receive modes respectively. It should be noted here that the time taken for the WUB transmission and for the data transmission is different (because of different packet size), and hence the energy consumption is different even though the level of the current drawn by the two is nearly the same.

Fig. 3.11 shows the effect of data collision. In this case, two transmitting nodes intend to transmit at the same time instant to the same destination node, both nodes find the channel as idle and transmit their data, which results in data collision over the channel. In this specific example, both transmitters, after sending their data, switch to

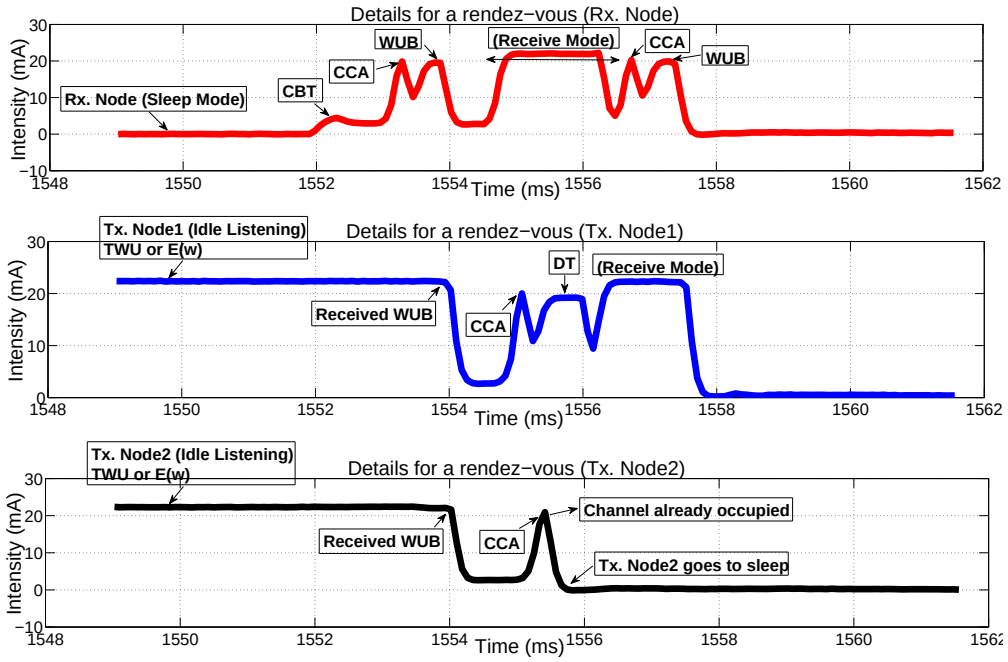


FIGURE 3.9: Same as Fig. 3.8, only three nodes are presented separately for clarity.

the receive mode and wait for the ACK signal from the receiver. Due to data collision over the channel, the receive node keeps waiting until $4ms$, which is three times more than the time during normal successful communication. Similarly, both transmitters wait more than $3ms$ for the ACK signal. The extra cost of this data collision at the two transmitters is found to be $0.3699.10^{-3}J$ and $0.3701.10^{-3}J$ respectively, whereas at the receiver, the energy is calculated as $0.3888.10^{-3}J$.

3.4 Scenario-Based Hybrid Energy Model

One of the important contribution in this chapter is concerned with more precise energy consumption estimation of the communication between nodes which consists of both software (protocol stack) and hardware (physical platform) parts. Traditionally the total energy consumption E_{tot} is the sum of three terms: E_{algo} is the energy consumed by the microcontroller for the protocol layers and their control, E_{proc} is the part of the energy consumed by shaping of the signal and it involves all the analog and digital signal processing including the conversion from analog to digital and vice versa. Finally, E_{amp} is the additional part that depends on the output power and it is mainly due to the power amplifier.

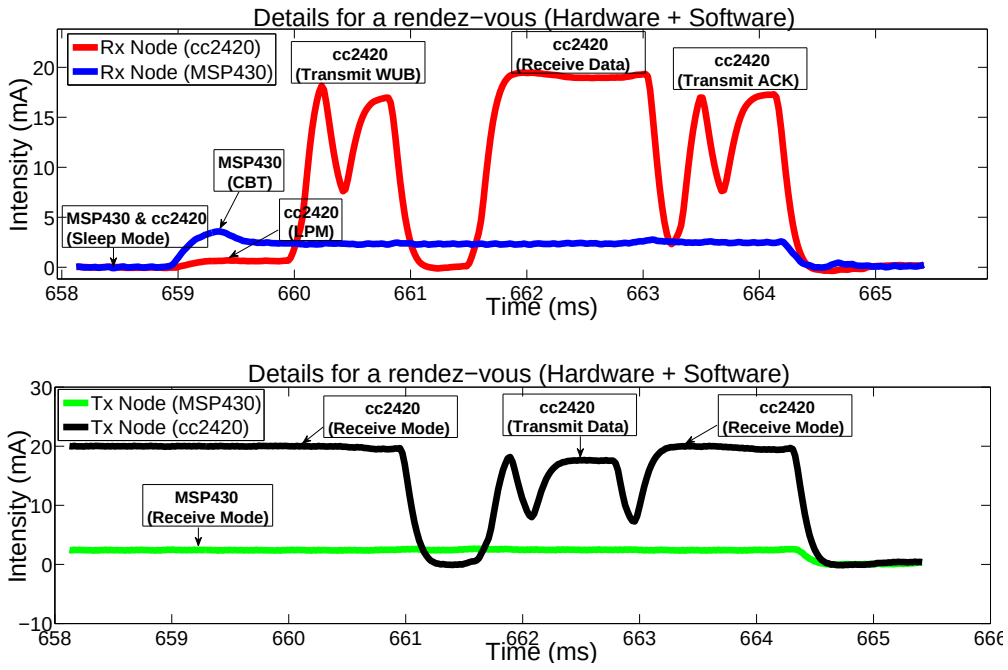


FIGURE 3.10: Details of software and hardware consumption: the software component i.e. $MSP430$ consumes 0.0 μA, 0.01 μA, 2.5 mA, and 3.7 mA, in sleep, low power mode (LPM), transmitting, and receiving modes respectively. Whereas, the hardware component i.e. $cc2420$ consumes 0.0 μA, 0.4 mA, 17.6 mA, and 19.6 mA, in sleep, low power mode (LPM), transmitting, and receiving modes respectively.

The evaluation of the term E_{algo} is done by measuring the processing time of the different tasks of the loop traces. The terms E_{proc} can be calculated by the power consumption of communication components in the receive or transmit modes, whereas E_{amp} depends on the transmit power level P_{Level}. Let P_r denote the power consumption of the signal processing components in receive mode, $P_t(P_{Level})$ and $P_t(min)$ the power consumption due to signal processing components in the transmit mode for an output power of P_{Level} and for a minimal transmission power respectively. T_r is the time spent in receive mode and T_t is the time spent in transmit mode, then E_{proc} and E_{amp} can be expressed as

$$E_{proc} = P_r.T_r + P_t(min).T_t \tag{3.31}$$

$$E_{amp} = [P_t(P_{Level}) - P_t(min)].T_t \tag{3.32}$$

This classical energy consumption model is only based on the hardware components, similarly there are purely software energy models such as [76] and [109]. These energy models do not give a precise and detailed measure of the energy because there are different scenarios which are associated with the software consumption and the hardware consumption. Therefore, to achieve accurate measure, it is necessary to identify the energy consumption of the scenarios which correspond to the occurrence of the events and

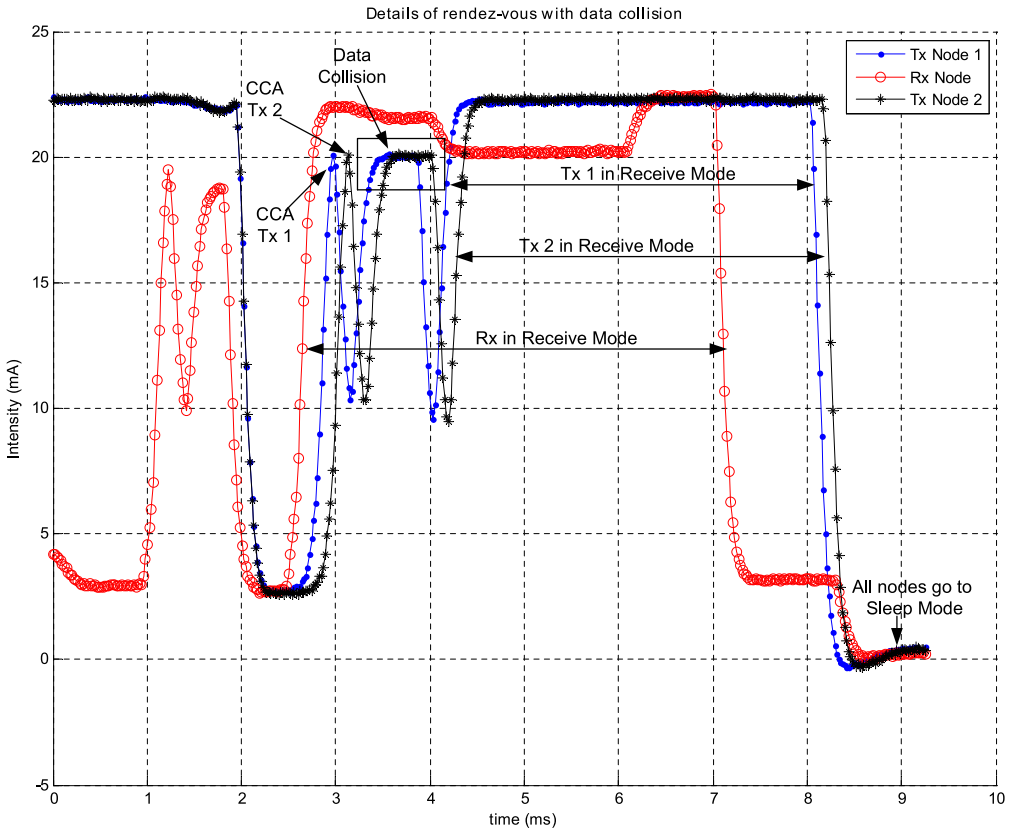


FIGURE 3.11: Unsuccessful communication because of data collision, which is one of the bad executing scenario as mentioned in the energy model. This results in extra power consumption because of the waiting of ACK for the transmit nodes and data packet for the receive node. After about 4 msec. of wait in monitoring state, all nodes go to sleep mode.

more precisely to their timings.

3.4.1 Scenario Descriptions and Cost Analyses

To introduce the scenario-based energy consumption, we consider an example of distributed asynchronous sensor networks (DASN). In DASN, a sensor node can wake-up at a particular instant and communicate with another node without any collision, but the same node can have a collision when it wakes up at another time instant. Consequently the energy consumption for the same node is completely different. Further, in the software, there is different energy consumption for being in the transmit or receive states, depending upon the software executing traces of different scenarios. The sleeping mode of the software components are linked directly with the frequency of the events (depending on I_{wu} and T_{wu}). If there are more events, the node does not go to deep

sleep mode or otherwise it can be in the deep sleep mode for multiples of I_{wu}. Similarly for the hardware consumption, for example, while transmitting the data or control packets, there are two distinct executions: the channel sensing (CCA) and the actual transmission, and their power consumptions are not the same. Therefore, we have to isolate them and not evaluate them through only one state (i.e. transmit state) as in classical models. We believe that by classifying the energy consumed by the software and hardware through scenarios provides much more details of the energy consumption and also the impact of cross layer on the energy consumption is more clear.

The executing scenarios which typically occur during a communication process are shown in Fig. 3.12. These scenarios are identified in the form of traces and called as, calculation before transmission (CBT), transmitter wake-up (TWU), wake-up-beacon (WUB), wake-up-collision (WUC), Data Transmission (DT), Data Receive (DR), Acknowledgment (ACK), Data Received With Errors (DRE) and *Data Collision* (DC). The scenarios which are shown in plain lines are common to every communication while the ones in dotted lines illustrate some bad events that may occur especially in asynchronous rendez-vous schemes.

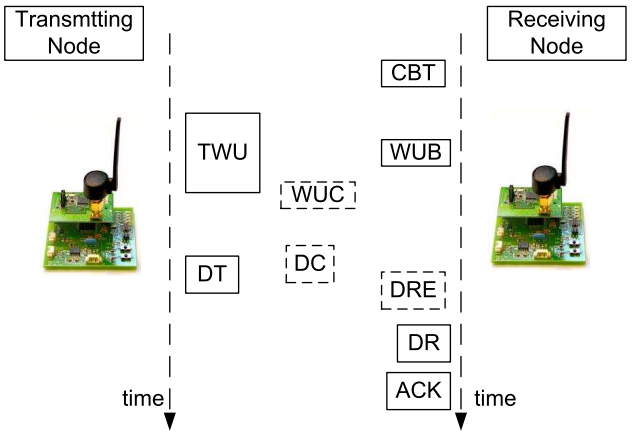


FIGURE 3.12: Communication phase between transmit *nodes* and receive *nodes*: The energy model is based on execution traces of different scenarios which occur during the communication. These scenarios include: CBT (calculation before transmission), TWU (transmitter wake-up), WUB (wake-up-beacon), WUC (wake-up-collision), DT (data transmission), DR (data received), DC (data collision), DRE (data received with errors) and ACK (acknowledgment of data).

Different costs are calculated based on the scenarios that have been identified during the communication and are presented below. The total energy cost is the sum of the costs of the main loop that records the execution traces (as explained in Sec. 3.3.2).

3.4.1.1 Basic Cost

The basic cost C_B is the cost of an ideal successful communication of one packet between two nodes, it is represented with plain line style through scenarios in Fig. 3.12 and is calculated as

$$C_B = C_{TWU} + 2.C_{CBT} + C_{DT} + C_{DR} + 2.C_{WUB} + 2.C_{ACK} \tag{3.33}$$

3.4.1.2 Re-transmission Cost

A re-transmission happens to be necessary when a received data frame is still erroneous after the error correction step, forcing the transmitter to realize another complete rendez-vous with the receiver. The additional cost of this event is

$$C_{RT} = C_{DRE} + C_B \tag{3.34}$$

where

$$C_{DRE} = C_{TWU} + 2.C_{CBT} + 2.C_{WUB} + C_{DT} + C_{DR}. \tag{3.35}$$

3.4.1.3 Data Collision Cost

Two or more transmitters may want to respond at the same time to the node waking up (i.e. the node that has broadcasted the wake-up-beacon), which might result in data collision. This bad situation wastes the energy of the transmitter, and causes an additional cost of C_{DC}. It is worth mentioning that C_{DC} is not straightforward to calculate since the behavior of the nodes during collision is unpredictable. For example according to our observations, the data collision results in the loss of packet identifiers and in that case the receive node keeps receiving the data packet for more than the duration of one packet. Therefore, C_{DC} depends on the actual effect of collision that can vary from each other. As an example, one of the behavior of two transmit nodes and one receive node during real-time data collision is shown in Fig. 3.11 and explained in Sec. 3.3.2.

3.4.1.4 Wake-Up-Collision Cost

If two receivers wake-up exactly at the same time and sense the channel being idle, they will transmit their wake-up-beacon at the same time which then results in wake-up-collision. This collision causes a waste of energy in the form of C_{WUC} at the receiver and it has a direct impact on C_{TWU} as the transmitting node has to keep waiting for longer time, and the term $\mathbb{E}[w]$ will therefore increase.

3.4.1.5 Total Energy Cost

The total energy cost of the communication from node i to node j is then calculated as (each term depending on i and j)

$$C_{Tot} = V_{i,j}.(C_B + N_{RT}.C_{RT} + N_{WUC}.C_{WUC} + N_{DC}.C_{DC}), \tag{3.36}$$

where $V_{i,j}$ is the volume of data (packet/s) transmitted between node i and node j, N_{RT} is the average number of re-transmissions per communication, N_{WUC} is the average number of wake-up-collisions per communication and N_{DC} is the average number of *data collisions* per communication. These values can be evaluated by counting the events on real platforms or by the following theoretical formulas (each term depending on i and j).

$$N_{RT} = \left(\frac{1}{1 - RT(tol)}\right) - 1 \tag{3.37}$$

$$N_{WUC} = \left(\frac{1}{1 - Pr_{WUC}}\right) - 1 \tag{3.38}$$

$$N_{DC} = \left(\frac{1}{1 - Pr_{DC}}\right) - 1 \tag{3.39}$$

where $RT(tol)$, Pr_{WUC} and Pr_{DC} are the re-transmission rate, probabilities of wake-up-collision, and data collision respectively. These terms can be evaluated through Eq. 3.2, Eq. 3.6 and Eq. 3.9.

3.4.2 Hybrid Energy Model

The hybrid energy model is presented in Fig 3.13. The model has input parameters *wake-up-interval* (I_{wu}), *wake-up time* (T_{wu}), *observation time* (T_{obs}), *amount of data packets* (λ) and *number of nodes in the radio range* (n_r). The model utilizes the real-time power measurements with an identification of typical scenarios that can occur in

asynchronous WSN communication as explained in Sec. 3.4.1. on the other hand, the analytical model evaluates the expressions for *number of re-transmissions* N_{RT}, *number of wake-up-collisions* N_{WUC} and *number of data collisions* N_{DC}. From these parameters together with traffic load (λ) the global cost and therefore the total energy consumption can be computed.

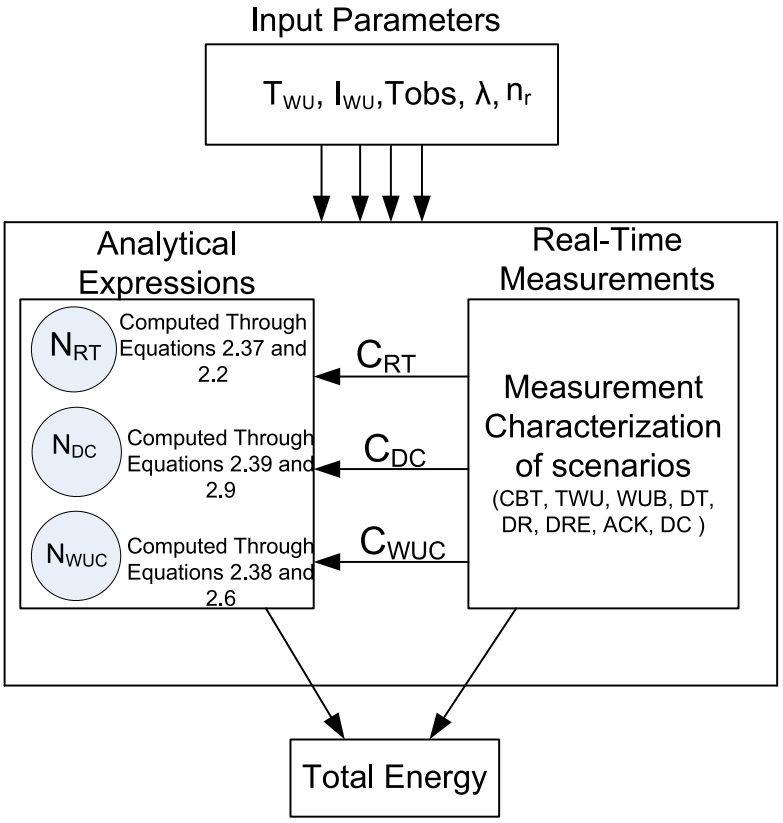


FIGURE 3.13: Hybrid Energy Model. The model utilizes the real-time measurements with an identification of typical scenarios that can occur in asynchronous WSN communication. By the help of the analytical model which evaluates the expressions for number of re-transmissions, number of wake-up-collisions and number of data collisions, the total global cost and the total energy consumption is evaluated.

3.4.3 Energy Consumption Measurements

Energy consumptions have been measured for different executing scenarios (as presented in Fig. 3.12) that can occur during the communication between sensor nodes. Tab. 3.2 illustrates the precise energy consumption of these scenarios which are identified in the form of executing traces. The consumptions have been isolated into software and hardware. For the case of software, the energy is measured through profiling of the actual code of different scenarios, whereas for the hardware, it is calculated based on the real-time measurements of the current consumption (from Fig. 3.8 and Fig. 3.10).

The current drawn in different states of the radio is interpreted in terms of numerical sample values and the integrated sum over the time results in the energy for different scenarios. The total energy consumption for one complete communication (i.e. for transmitting and receiving one packet between two nodes which are within the radio-range) includes all the energy required for scenarios that are shown in Tab. 3.2, and also the energy of TWU. The energy consumption of the transmitting node in Joules (J) is $(0.0483,\ 0.0052)J$ for ($cc2420$, $MSP430$), whereas for the receive node it is calculated as $(0.1789,\ 0.0043)\times 10^{-3}J$. The transmitter consumption is 10^3 times more in comparison to the receiver because of the penalty of long waiting time $\mathbb{E}[w]$ for the WUB. The energy consumption at the transmitter due to TWU is calculated as $0.0431J$, which is nearly equal to the total energy consumed by the transmitting node during one complete communication or more precisely 10^3 times more in comparison to the rest of the consumption for the transmitting node. This concludes that the main bottleneck (in the selected MAC protocol of preamble sampling category) lies in expected wait $\mathbb{E}[w]$ for the wake-up beacon. Having said that, this consumption can significantly be reduced by applying adaptive wake-up intervals at each node according to the data traffic.

TABLE 3.2: Software and hardware energy consumptions (Joules) of the executing scenarios which are identified during communication between sensor nodes. The energy consumption, in different states (tx, rx, $link$, and $ntwk$), and of the associated components $timer$ and $controller$ of the software component (microprocessor $MSP430$) are presented. For the case of hardware component (radio transceiver $cc2420$), energy consumption in transmit ($cc.Tx$) and receive ($cc.Rx$) states are presented.

SW/HW	CBT	WUB/ACK	DT	WUC	DC
tx	$2.6.10^{-8}$	$5.3.10^{-8}$	$1.2.10^{-5}$	$4.1.10^{-8}$	$1.0.10^{-5}$
rx	$5.3.10^{-8}$	$1.2.10^{-6}$	$5.3.10^{-8}$	$1.3.10^{-6}$	$5.3.10^{-8}$
$timing$	$5.5.10^{-7}$	$5.5.10^{-7}$	$5.5.10^{-7}$	$5.5.10^{-7}$	$5.5.10^{-7}$
$link$	$4.8.10^{-7}$	0	0	0	$4.8.10^{-7}$
$ntwk$	$4.8.10^{-7}$	0	0	0	0
$contr.$	$3.7.10^{-7}$	$3.7.10^{-7}$	$3.7.10^{-7}$	$3.7.10^{-7}$	$3.7.10^{-7}$
$cc.Tx$	0	0	$6.8.10^{-5}$	0	$2.9.10^{-4}$
$cc.Rx$	$7.8.10^{-6}$	$5.0.10^{-5}$	$1.0.10^{-4}$	$3.1.10^{-5}$	$2.3.10^{-4}$
$Total$	$9.7.10^{-6}$	$5.1.10^{-5}$	$1.7.10^{-4}$	$3.2.10^{-5}$	$5.3.10^{-4}$

3.5 Validation and Performance Evaluation

In order to validate the presented energy model, a test scenario as shown in Fig. 3.14 is considered to compare the real-time energy consumption of the sensor nodes with the

estimated model. Let us consider two nodes A and B which are intending to transmit the data to the node C at a specific rate, the real-time energy consumption of node C is evaluated for a certain duration of time. This scenario can be applicable to a situation where the node C acts as a first hop between the node A or B. C will then forward the received data to the destination nodes F and G, which can be out of range from A and B. The real-time current measurements of the monitored node C are exported as a *.csv* file to the external memory of *Agilent* $N6705A$ DC power analyzer to compute the average energy consumption. The observation time is one of the constraint in the validation of our model due to limited memory size of the power analyzer to keep the logged data, therefore the real time measurements are not possible for many hours or days and for large scale networks.

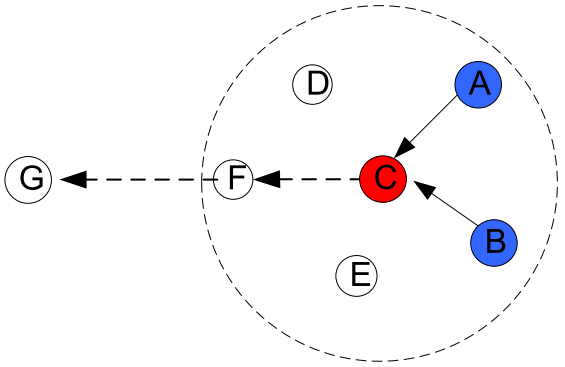


FIGURE 3.14: Two nodes A and B are intending to transmit the data to the node C at a specific rate, and the energy consumption of node C is monitored.

Three different experiments are presented in Fig. 3.15 for the comparison of the estimation model and the real-time measurements. In order to have a reliable measure of the estimation model, the real-time measurements have been repeated five time (i.e. M1 to M5), for every test with the same setting of parameters. During these tests the parameters considered fixed are I_{wu}, T_{wu}, T_{obs}, and TWU with respective values of $100ms$, $5ms$, $300s$ and $2s$. The data transmission rate from node A and B to the monitored node C varies in three tests from $1s$, $500ms$ and $100ms$ for test 1, test 2 and test 3 respectively. For the case of real-time measurements, the average energy per second is computed from $300s$ of observation time which contains 2.9×10^6 samples of the current levels for a sampling resolution of $0.1ms$. The results shown in Fig. 3.15 extend for $300s$ of observation time T_{obs}. For the estimated model, the energy consumption of the scenarios shown in Tab. 3.2 is injected in the model to compute the average energy per second and further for $300s$ which is compared with the actual measurements as shown in Fig. 3.15.

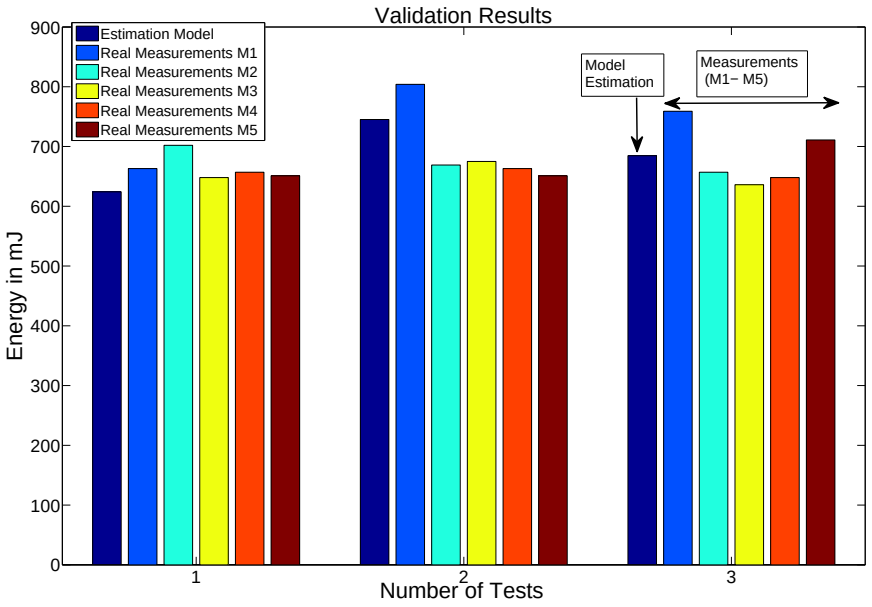


FIGURE 3.15: Test 1, Test 2 and Test 3 correspond to fixed parameters settings of I_{wu}, T_{wu}, T_{obs}, and TWU as $100ms$, $5ms$, $300s$, $2s$, whereas the data reception at node C varies from $1s$, $500ms$ and $100ms$. Y-Axis represents the energy consumed in mJ for an observation time of $300s$.

The energy consumption of estimated model versus real time measurements (averaged) for test 1, test 2 and test 3 are shown in Tab. 3.3. Results verify that the relative error of our energy model is between 1% to 8%. As the estimated model uses the energy consumption of the scenarios which are evaluated through real-time measurements, therefore we have very accurate energy comparison. The validation test is being performed at the receive node and we considered the energy consumption of wake-up beacon and the reception of data packet.

TABLE 3.3: Performance evaluation of validated results.

	Test 1	Test 2	Test 3
Model Estimate (mJ)	624.2	745.2	684.6
Average Measurements (mJ)	664.2	692.2	682.2
Standard Deviation	19.59	56.36	46.18
Relative Error	6.0%	7.5%	0.35%

The deviation of the energy model depends upon the parameter *wake-up time* (T_{wu}) of the receive node. In the real-time experiments, the receive node, which is a monitored node C of our validation scenario has a variation in waiting for the data packet between 1.5 to 5 ms. This variation is due to the exact wake-up of the transmit nodes. If the transmit node wakes up earlier, the waiting time at the receive node will be less, which consequently will consume less energy, but $5ms$ is the maximum *wake-up time* (T_{wu})

that the receive node waits for the data. Whereas in our estimate model, the waiting time is taken as an average value between 1.5 to 5ms.

3.6 Applications of the Energy Model

In this section we exploited the energy model and presented few results. Moreover, later in this thesis the model is further utilized for accurate energy consumption evaluation at the MAC and PHY layers.

3.6.1 Effects of Collisions and Total Energy

In order to illustrate the effects of collisions on the overall power consumption of the individual nodes and consequently of the network, a numerical analysis is presented in Tab. 3.4, which evaluates the extra waste of energy due to collisions. The amount of collisions in a network is calculated through analytical expressions represented in Sec. 3.2.2.1, whereas the energy wasted due to collisions is calculated through the results of real-time measurements. For the numerical analysis, three different I_{wu} values and variable traffic rates of data packets are considered. The probability of WUC and DC have been obtained through Eq. 3.6 and Eq. 3.9 respectively, where the number of nodes within the radio range is taken as n_r = 10, T_{wu} = $5ms$ is the time that the receive node waits for the data packet from transmit node after sending its WUB and T_{obs} = $1hr$ is the network observation time. To support various WSN applications the data packets are generated with rate $(1/3, 1/2, 1)$ times of I_{wu} $(3, 0.3, 0.1)s$, whereas the WUB is transmitted at every I_{wu}.

It is observed that the probability of collisions increases sharply when a shorter I_{wu} is selected to support heavy traffic. This comes with a significant increment in the actual energy consumption. Obviously, the selection of I_{wu} depends upon the specific application. For typical WSN applications such as critical temperature sensing or environment monitoring, it is evaluated that the probabilities of WUC and DC have resulted in 21.5% increase of the energy consumption in comparison to the energy consumed during transmission and reception with no collisions. The traffic and I_{wu} have a direct impact on the amount of collisions. It is worth mentioning here that, even if there is no collision (i.e. non-contention based protocol), our energy model and the real-time evaluations are

still more accurate than the basic power models which consider constant current levels in different states such as Tx, Rx, etc.

TABLE 3.4: The cost of collision energy (C.Energy) results in an extra power consumption of the individual nodes and consequently of the complete network. The traffic is generated for different I_{wu} and accordingly the probability of collisions are calculated to compute the collision energy.

I_{wu}	Packets/hr	WUB/hr	WUC	DC	C. Energy
$3s$	400	1200	1.5%	2.2%	$23.8mJ$
$.3s$	6000	12000	15%	4.6%	$1107\ mJ$
$.1s$	36000	36000	45%	11.2%	$17115mJ$

3.6.2 Optimization of the Output Transmit Power

Fig. 3.16 illustrates one of the potential application of our energy model. The total energy per successfully transmitted useful bit between nodes i and j is calculated as a function of the distance $D_{i,j}$ and of the transmission power level P_{Level}. Now, for given i and j, a particular P_{Level} allows to optimize the total power. The optimal point in terms of energy consumption can be found on the curve which gives the best relationship between P_{Level} and $D_{i,j}$. The dynamic adaptation of the transmit power is further explored later in this thesis. Similar examples can be obtained such as for different error correcting codes we can determine the optimal point on the curve against the power consumption and the energy per useful transmitted bit.

The life-time prediction of the system can be evaluated by considering a geographical network and by using a routing algorithm to compute the volume of traffic for all the nodes of the network and thereafter by using our energy model we can predict the life time of the network.

3.7 Conclusion

In this chapter a hybrid and scenario-based energy model is presented for accurate energy consumption analysis of a WSN. Firstly, an analytical model has been described for the lower layers of protocol stack such as PHY, MAC, DLL, layers, to have a global overview of the energy utilization in the system. The MAC strategy has a particular impact on the performance of the whole system, since the consumption of the radio part is the most

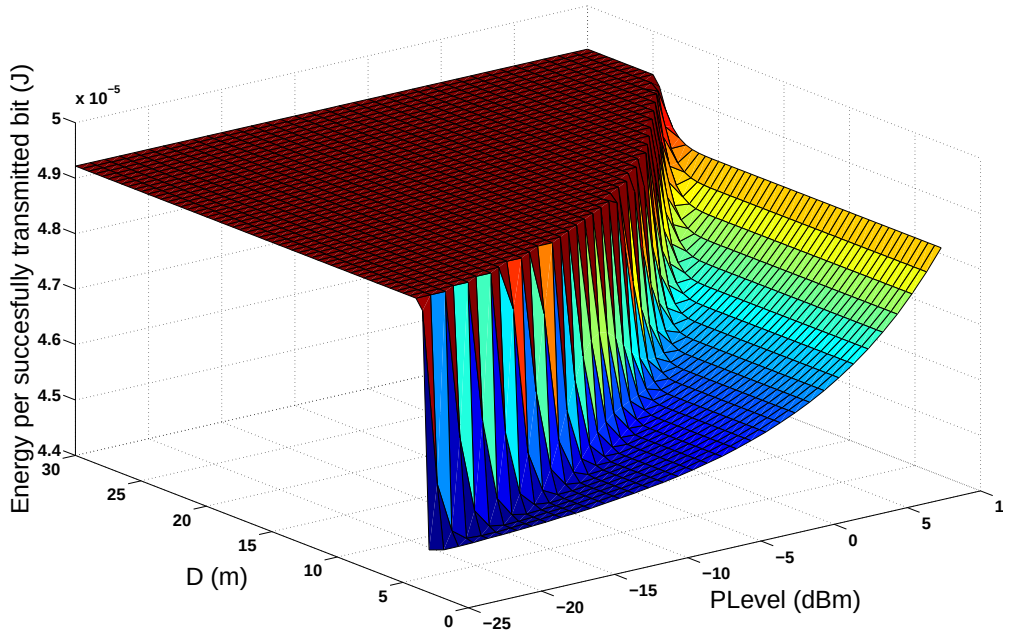


FIGURE 3.16: Total energy per successfully transmitted useful bit between nodes i and j as a function of the distance $D_{i,j}$ and of the transmission power level P_{Level}.

important. That is why, the MAC layer has been precisely modeled for preamble sampling category with special focus on receiver initiated cycled receiver (RICER) protocol. Then the real-time measurements have been presented to understand the details of the energy consumption by isolating the hardware and software consumption. The detailed current consumption of the components in various states were evaluated. Further, the details of the rendez-vous including the collision avoidance and data collision have been presented.

A realistic and accurate energy model for WSN has been proposed. The power evaluation was done by the analysis of the different scenarios that may occur during the communication between sensor nodes. These scenarios have been isolated, characterized, and injected into the energy consumption model. The proposed energy model can help system architects to identify the critical regions for energy savings with a better understanding of the variations in the current levels that occur in different scenarios. Furthermore, it enables the designers to achieve more accurate and energy efficient designs than using classical approaches for energy modeling.

The presented model was validated under different test cases which shows that the relative error of the estimated model is less than 8% in comparison with the real-time energy

estimate. At the end, some examples have been given to explain which are the applications of our energy model. The model presented is specific to RICER protocol, however it gives details of the power consumption based on the scenarios which can be identified in other energy efficient protocols. Therefore, in a larger context our model can easily be extended to other preamble sampling categories of MAC protocols. In the next chapter, *adaptive wake up intervals* in the MAC protocol will be introduced to reduce the *cost of monitoring* ($\mathbb{E}[w]$) at the transmit node as well as the cost of idle channel monitoring of the receive node.

Chapter 4

TAD-MAC: Traffic-Aware Dynamic MAC Protocol for Wireless Sensor Networks

In the context of global energy consumption, MAC layer plays a pivotal role in improving the WSN lifetime by efficiently sharing the medium among the nodes in a network. It is concluded in the previous chapter that in distributed asynchronous sensor networks collisions and idle listening are the main source of energy consumption. Moreover, it is recommended that adapting the wake-up interval based on the traffic characteristics can help to significantly reduce energy consumption at the MAC layer for WSN. In this chapter a novel traffic-aware dynamic MAC (TAD-MAC) protocol is presented with an objective to minimize the energy consumption for WSN. The protocol is applied (later in Chapter 5) in the context of wireless body area sensor networks (WBASN). It can be considered as an addition in the class of preamble sampling MAC protocols [19] for variable traffic and application dynamics that can occur in WSN.

4.1 Introduction

Self-organization is extremely vital in resource constrained systems such as wireless sensor networks (WSN). Generally, natural phenomenon and design strategy are two views on the self-organization in any system [84]. Natural phenomenon of self-organization is

like a distributed system, where components interact on the micro level and organize themselves in a way that leads to macro organization. Whereas in design strategy, system components have to follow certain local rules intended to reach a desired global behavior. Typically, WSN nodes have limited energy source, constraints on the miniature size, low signal processing capability and low storage capacity. Since energy consumption remains the most important issue in WSN, in this chapter we present an approach towards self-organization focused on local energy optimization at the MAC layer which can lead towards network level global energy optimization.

In a typical WSN platform, the radio transceiver consumes most of the power. Radio activity is controlled by the medium access control (MAC) layer, therefore it is necessary to design an ultra low power and energy efficient MAC protocol suitable for WSN. In this regard, low duty-cycle protocols such as preamble sampling MAC protocols are very efficient because these protocols improve the lifetime of the network by reducing the unnecessary energy waste [19], but their performance in terms of energy efficiency is questionable under variable traffic environment. Number of different static low-duty cycle MAC protocols exist in the literature but to best of our knowledge there is no dynamic duty cycle protocol specific to traffic variations.

In this chapter a novel traffic-aware dynamic MAC (TAD-MAC) protocol is proposed. In our proposed adaptive technique, every node adapts its wake-up interval (I_{wu}) dynamically with due account of the amount of traffic (i.e. data packets) it receives and consequently optimizes the energy consumption. The traffic is estimated by using a traffic status register bank (TSR-bank) which contains the traffic statistics to continuously update the I_{wu} of the receive node with respect to the data transmission rate of the transmit nodes.

Further, in this chapter a detailed analysis on the convergence and performance of TAD-MAC protocol is presented. An adaptive system is used to adjust the best wake-up schedule of all the sensor nodes in the network. This system is integrated with a sensing environment (i.e. wireless sensor networks) which includes all the nodes, channel propagation delay, system latencies and hardware delays. The objective is to understand the convergence behavior of the sensor nodes under the variations of the sensing environment and traffic variations due to geographical position of nodes within a network and

application variations. In this regard, the system is modeled as open and closed-loop forms, and parameters are individually characterized.

The TAD-MAC protocol is implemented in an event-driven network simulator (WSNet). The performance metrics, i.e. energy consumption, latency, delivery ratio, and convergence speed, are calculated (for reaching a steady state) for different traffic rates through a small-scale single-hop network. A small-scale network is used as the primary focus to show how many data packets a node has to transmit before it can be synchronized with its neighbors. Results show that I_{wu} converges to a steady state value after a limited number of wake-ups. Further, best, worst, and average convergence times are presented. Finally, traffic characteristics are exploited from the network simulator and, by utilizing the energy model presented in previous chapter, precise energy consumption is evaluated for both transmit and receive nodes.

The rest of this chapter is organized as follows. In Sec. 4.2, energy efficient MAC protocols design principles are presented followed by the related work in Sec. 4.3. TAD-MAC protocol is described in Sec. 4.4 and then the system characterization of the TAD-MAC protocol is presented in Sec. 4.5. Implementation and performance evaluation are explained in Sec. 4.6 and finally, the chapter ends with the conclusion.

4.2 MAC Protocols Design Principles

The main objective of the MAC layer is to keep the energy consumption low by turning off the radio module as often as possible. In the context of energy efficient MAC protocols for WSN, the preamble sampling method (explained in Sec. 2.2) is an attractive option [19]. This section explains different design aspects of the MAC layer (from the point of view of energy consumption) after being able to precisely model the energy consumption of the sensor node (presented in chapter 3). Therefore, it is different than Sec. 2.2 which explains general MAC layer principles.

Idle listening, overhearing, overheads, unnecessary beacon transmissions and collisions are the major sources of energy waste in preamble sampling MAC protocols of WSN/WBASN . Idle listening (in receiver initiated protocols) occur when a sensor node is waiting for the receive node to wake-up (and to send the beacon) such that the transmit

node transmits its data as reflected by receive node initiated communication mechanism in Fig. 2.2. In the receive node initiated communication, it is the transmit node which wastes the energy due to idle listening. Moreover, the possibility of two nodes transmitting to the same receive node at the same time will result in collision. In this regard, as shown in previous chapter (Sec. 3.4.2), transmit node energy consumption can reach 100 times more in comparison with receive node. One of the advantage of using receiver initiated mechanism is that there is no overhearing cost as the transmit node sends data to the specific destination on the reception of a beacon. Moreover, this way of communication reduces the long preambles and uses less overheads.

On the other hand, if the communication is initiated by the transmitter, it appends long preambles with the data, which introduces no idle listening at the start before transmitting but it has to wait for a long duration in idle listening for the response (in the form of the acknowledgment) from the receiver. It results in a significant energy waste due to overhearing as all the neighboring nodes receive the preambles and the cost of transmitting long preambles (overheads) is also very high as for BMAC. This style of communication also impacts on the channel occupation and results in long delays in transmission.

To conclude, if the transmit node adapts its transmission time according to the wake-up schedule of the receive node (as it is the case in WiseMAC [60]), the transmit node can reduce idle listening and long preamble transmissions. WiseMAC is effective in static network and for fixed traffic, but in the case of variable traffic the receive nodes wake-up schedule should also be adaptable and there is a need of modification in WiseMAC. On the other hand, in the case of receiver node initiated communications, idle listening and collisions can be avoided by introducing adaptive wake-up interval as proposed in this chapter.

4.3 State of the Art

In the context of energy efficient MAC protocols, three classes of MAC protocols are widely used in WSN: the preamble sampling protocols, protocols with common active period and scheduled protocols. The channel polling approach including low power listening or cycled receiver protocols are part of a broader class of protocols called preamble

sampling protocols [19]. These protocols are most popular in distributed asynchronous wireless sensor networks because there is no requirement of synchronization and they are easily adaptable to various applications. The preamble sampling category (as explained in Sec. 2.2) includes energy-efficient MAC protocols which reduces the cost of extra overheads (in comparison with schedule-based protocols) and synchronizations by having single or multiple preambles. In addition, a protocol such as WiseMAC can efficiently adjust the duty cycle based on the 'wake-up time' of the neighbor nodes, which results in a great reduction of idle monitoring, preamble size and probability of collisions [60]. On the other hand protocols with common active period such as S-MAC [126], PMAC [130] and T-MAC [37] introduce predefined fixed and variable periodic listening and sleep schedule to avoid collision, overhearing and also reduce idle listening. Finally, schedule based protocols such as *IEEE 802.15.6* [2], SS-TDMA [73], PicoRadio [100], etc., achieve energy efficiency by having contention free time slots at an extra cost of synchronization.

It is necessary that the MAC layer incorporates in its design phase, various application dynamics that create traffic variations. A key point is therefore that the MAC layer can adapt itself to various environments, in order to optimize energy consumption. WiseMAC [60] can be considered as the best MAC protocol in the class of preamble sampling, but its performance degrades significantly under variable traffic load. [62] presents an extended WiseMAC to improve the throughput but it does not address the issue of fixed wake-up interval which is the main problem under variable traffic load. Few other MAC protocols such as [114], [118], and [117] are low-energy MAC protocols that optimize the energy waste due to idle listening, collisions and interference. [118] proposed a predictive wakeup (PW-MAC) protocol in which each node wakeup time is calculated through a random generator. All the nodes have knowledge of the initial *seed* value of their neighbor nodes. It is an efficient technique for static network but it requires re-configuration as soon as the traffic changes to broadcast the new *seed* value to the neighbors. Moreover, the technique has fundamental limitations due to the fact that the random generator is not a precise measure of packet generation. For example, a source node may have a data packet to transmit, but it will delay its transmission until the wakeup time of the receive node, which results in a transmission delay. Furthermore, the protocol is completely distributed and in asynchronous sensor networks there is always certain probability of wakeup and data collisions [14], which will sharply degrade the performance of PW-MAC.

In the context of self-organized preamble sampling MAC protocols, very few papers exist in the literature that propose to adapt preambles according to traffic variations. In this regard, TAD-MAC adds an important dynamic protocol in the category of preamble sampling MAC protocols. In this chapter the behavior of the TAD-MAC protocol is characterized and validated in terms of convergence to a steady state through rigorous testing.

4.4 Traffic-Aware Dynamic MAC Protocol

Wake-up interval and wake-up time are the most important design parameters in the preamble sampling class of MAC protocols. For periodic sensing and variable sensing applications, the wake-up interval is usually kept fixed, which results in degrading the performance as well as the energy efficiency. In this section a dynamic MAC protocol (called traffic-aware dynamic MAC, i.e., TAD-MAC) is proposed to adapt the wake-up schedule based on the estimation of the traffic load. The proposed approach can further be applied to other low power asynchronous MAC protocols (with some modifications as for instance explained in Sec. 5.4.2) such as B-MAC [95], X-MAC [24], WiseMAC [60], CyclED Receiver [82], RI-MAC [114] to reduce their energy consumption.

4.4.1 Protocol Description

The basic principle of the proposed protocol, which is initiated by the receive node, is described in Fig. 4.1. Fig. 4.1 shows an example of a small network which includes one receive node and three transmit nodes, and is a simplified version of a complete network. In normal circumstances, each node can be a transceiver and the network can be easily extended. The figure is divided into two phases (i.e. (a) before convergence and (b) after convergence), that sketch the results for three nodes (TxN1 to TxN3) attempting to transmit data to a receive node R. During the evolution phase (Fig. 4.1-a) before reaching a steady state, each transmit node (TxNi) waits for the beacon signal from the coordinator before sending its data. The beacon transmitted from the coordinator contains the specific node ID (identifier). Other intending transmit nodes continue to wait for receiving their beacons.

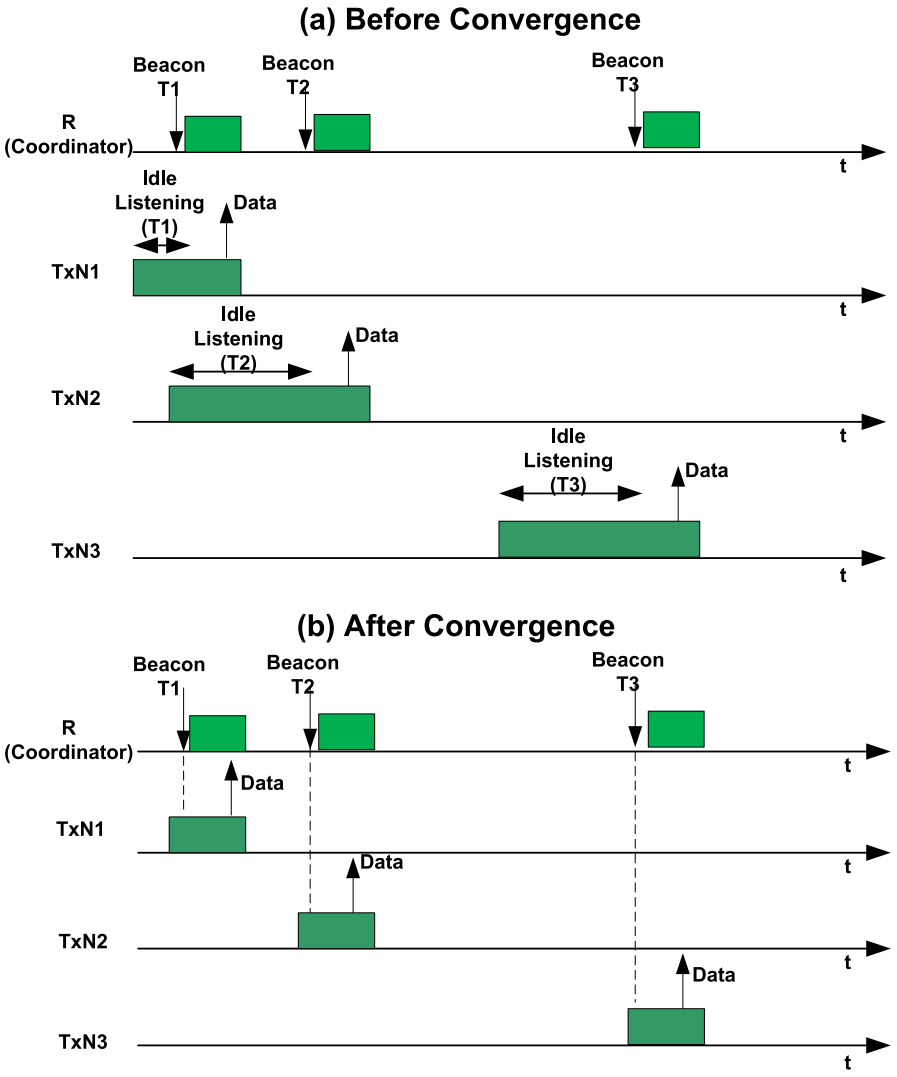


FIGURE 4.1: Traffic-aware dynamic MAC protocol is divided into two phases; the evolution phase before convergence (a) and the steady state phase after convergence (b). It is important to note that in the second phase the receive node (coordinator) has adapted its I_{wu} schedule as such that the idle listening is reduced.

After several wake-ups of the coordinating node for various transmit nodes, the coordinator will adapt its wake-up interval (I_{wu}) based on the statistics of the traffic it receives from each individual node. The second phase after the convergence (Fig. 4.1-b) indicates that the receive node (coordinator) has adapted its I_{wu} schedule in such a way that idle listening is reduced. In order to cater for the clock drift and hardware latencies, the receive node sends the wake-up beacon (WUB) slightly after its scheduled time to ensure that the intending transmit node is already awake.

It is important to mention the exact difference between TAD-MAC and other low-power MAC protocols such as WiseMAC. Basically, both protocols (in a distributed fashion) reduce idle listening by synchronizing the wake-up schedule. One of the fundamental

difference is that, WiseMAC is based on a fixed wake-up interval which is learned (from receive node to transmit node) after every communication between the two nodes but this wake-up interval (I_{WU} mentioned in Fig. 2.5) remains fixed irrespective of variation in the traffic between different sensor nodes. Whereas, TAD-MAC is based on an evolutionary process of evaluating the wake-up schedule of the sensor nodes through an estimation of traffic load and the process is adaptable to variable traffic load. Having said that, WiseMAC is used for down-link of infrastructure network (from access point to a sensor node), which is a low traffic as it only contains the configuration and querying requests [59]. However, in the case of variable and high traffic the performance of WiseMAC degrades rapidly [118]. Whereas, the focus of TAD-MAC is to remain generic and applicable to variable traffic loads to meet the demands of various WSN applications.

The main novelty of the proposed TAD-MAC protocol exist in dynamic adaptation of wake-up interval under variable traffic. There are two fundamental parameters (i.e. wake-up interval and wake-up time) that are required to achieve synchronization between transmit and receive nodes. Fig. 4.2 shows the basic block diagram of the TAD-MAC protocol. Since TAD-MAC is based on receiver-initiated mechanism, the receive node estimates the traffic load and based on this estimation it controls/adapts the wake-up interval in order to synchronize with the data transmission time of the transmit node.



FIGURE 4.2: Traffic-aware dynamic MAC protocol consists of two components (traffic estimation and adaptive wake-up interval algorithm).

4.4.2 Traffic Estimation

The traffic estimation technique has to be low in complexity (such that it can be executed fast) while remaining effective. In this regard, we introduce a traffic status register (TSR) that contains a status information of the traffic at each node. On the wake-up of a receive node it sends a wake-up beacon, if it receives a data in response, the TSR corresponding to the transmitted node is filled with a $'1'$ and if it does not receive data, the TSR is filled with a $'0'$. The new status (either 1 or 0) is filled in the 1st index of the register. The register contents are shifted one bit right before inserting a new status. This is a blind way of keeping the track of traffic load without having any information about the data

rate of the transmitter. However, this blind estimation is not a quantitative evaluation of the data rate but rather a relative measure of either the traffic is increasing (which means multiple consecutive ones in TSR) or decreasing (which means multiple zeros in TSR).

In this context, for the adaptation of I_{wu}, each node contains a traffic status register bank as shown in Fig. 4.3 to keep the traffic status of all the neighbor nodes. Each register corresponds to an individual neighbor node which is updated in response to the wake-up beacon (sent by the receive node to the transmit node) and the response (from the transmit node) can be either a data packet or no data packet. The receive node wakes up and sends its beacon (solicited) to the node which has the nearest wake-up time T_{wu}. The beacon packet contains the field about the specific node ID which should transmit its data.

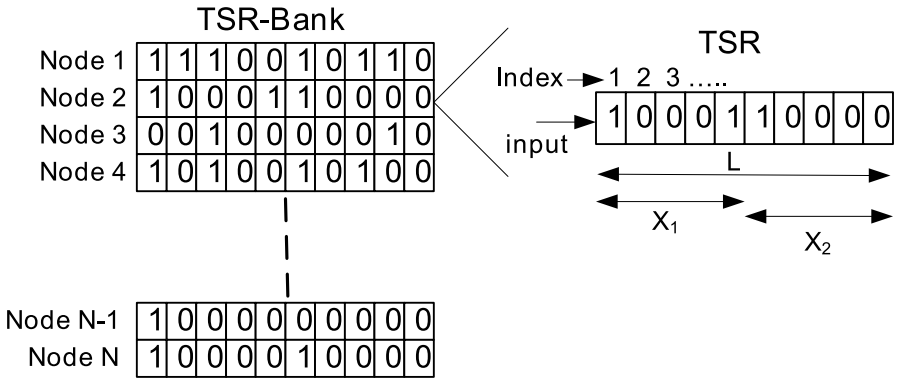


FIGURE 4.3: Traffic status register bank: it contains N registers for N neighbor nodes. If the receive node receives data, the register (of the corresponding transmit node) is filled with $'1'$ and if it does not receive data, the TSR is filled with $'0'$.

The wake-up interval for an individual node is estimated based on the contents of the traffic status register. The register is divided into two halves in order to consider a dominant impact from the most recent traffic (which resides in *most significant* part) in comparison to the relatively old traffic (which resides in the *least significant* part). The size of the register is a parameter that can be tuned to achieve fast convergence speed. The TSR can have different patterns depending upon the variations in the traffic and position of the node. Further it is used to evaluate the next wake-up interval and helps to reduce the unnecessary wake-up beacon or preamble transmissions.

Now, before going into the details of the adaptive algorithm it is important to mention the desired TSR pattern or a sequence of convergence that is targeted for TAD-MAC. The algorithm (presented in Sec. 4.4.3) is designed such that as it converges to a steady

state value of wake-up interval the TSR contains a sequence of [10101010...] pattern. This typical sequence seems the best trade-off between the optimal wake-up interval [11111111...] (i.e. when each wake-up of the node is followed with a successful data reception), and too frequent wake-up [100010001...] due to several reasons:

- Sequence of [11111111...] is an ideal sequence which means that each time the receive node wakes up it receives data but it always has probability of missing data as soon as the traffic rate increases (at the transmit side). The wake-up interval of the receive node will keep converged at lower traffic rate (irrespective of the fact that the traffic rate has increased at the transmit side) because of [11111111...] sequence but it will loose number of packets. Moreover, the packets keep on aggregating at the transmitter side which will degrade the performance both in terms of energy and delivery ratio.

- Whereas, if the traffic rate reduces then, [11111111...] sequence will become [01111111...] and consequently the wake-up interval will no more remain converged and the convergence process will start for a new traffic rate. So in the case of reducing the traffic rate there is no problem while selecting [11111111...] sequence because it can learn that the traffic has reduced and the wake-up interval should be reduced (this is also the case with other sequences).

- On the other hand, sequence of [10001000...] is another extreme as the receive node wakes up too often and results in unnecessary beacon transmission and energy wastage.

- [11101110...] sequence is another alternate which increases the capacity and can be more energy efficient but its response to an increase traffic is slow because it will take two more wake-ups (in worst-case) to be able to know that the traffic has increased. This is especially critical for the burst traffic when the response has to be extremely reactive.

- [10101010...] sequence seems to be the best trade-off between [11111111...], [11101110...] and [10001000...] sequences under the above considerations and is selected as a steady state in this work.

4.4.3 Adaptive Algorithm

As the node wakes up, it sends the wake-up beacon and then it computes the next wake-up interval based on the TSR. The wake-up interval is updated for the time instant $(t_i + 1)$ based on the values of I_{wu} and on two variables, μ and e at time instant (t_i) as

$$I_{wu}(t_{i+1}) = I_{wu}(t_i) + [\mu(t_i) + e(t_i)] \cdot t_{ref}. \tag{4.1}$$

Where, μ is the output of the weighting average algorithm and e is the correlation error. In order to have a unique reference of time among I_{wu}, μ, and e, t_{ref} is multiplied with μ and e, and it is defined with reference to the system/simulator clock. The update factor $\mu(t_i)$ is calculated based on the computation of two weighted values i.e., most significant $(X_1(t_i))$ and least significant $(X_2(t_i))$ (shown in Fig. 4.3) as

$$\mu(t_i) = \alpha \cdot X_1(t_i) + (1 - \alpha) \cdot X_2(t_i), \tag{4.2}$$

where α is a constant weighting factor. $X_1(t_i)$ and $X_2(t_i)$ are defined as

$$X_1(t_i) = \frac{N_{0,1}}{L/2} \cdot Nc_{0,1} - \frac{N_{1,1}}{L/2} \cdot Nc_{1,1}, \tag{4.3}$$

$$X_2(t_i) = \frac{N_{0,2}}{L/2} \cdot Nc_{0,2} - \frac{N_{1,2}}{L/2} \cdot Nc_{1,2}, \tag{4.4}$$

where $N_{0,1}$, $N_{1,1}$, $Nc_{0,1}$, and $Nc_{1,1}$ are the number of zeros, number of ones, number of pairs of consecutive zeros and number of pairs of consecutive ones in X_1 respectively, whereas, $N_{0,2}$, $N_{1,2}$, $Nc_{0,2}$, and $Nc_{1,2}$ are the number of zeros, number of ones, occurrence of consecutive zeros and the occurrence of consecutive ones in X_2 respectively. For example a sequence [11100100] has $Nc_{1,1}$ equals to 2 and $Nc_{0,1}$ equals to 0, whereas, $Nc_{1,2}$ is equal to 0 and $Nc_{0,2}$ equal to 1. The optimization parameter L is the length of the TSR, which can be tuned to achieve fast convergence in the context of WBASN as explained in Sec. 5.3.

The adaptive wake-up interval system is illustrated in Fig. 4.4. After the weighting average algorithm computation of the wake-up interval $\mu(t_i)$ through (4.2) to (4.4), a

cross correlator is used to smooth the output of the weighting average algorithm. The correlator is fed with the current contents of TSR and it computes a correlation after a comparison with the reference patterns that is [10101010...]. The correlator provides the output error either positive (which means that the TSR sequence contains more zeros than ones, hence, the correlator output should contribute to increase the wake-up interval) or negative (which means that the TSR sequence contains more ones than zeros, hence, the correlator output should contribute to decrease the wake-up interval). Consequently it guides the adaptation of wake-up interval towards the desired sequence. Finally, the previous wake-up interval value is added with $\mu(t_i)$ and $e(t_i)$ to compute the next wake-up interval at the output according to Eq. 4.1.

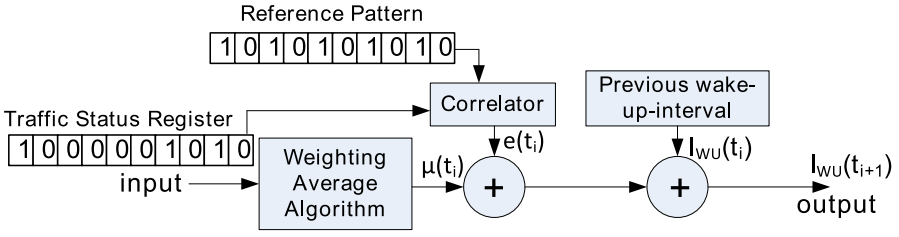


FIGURE 4.4: Adaptive wake-up interval system.

In the proposed algorithm the most influential parameter is Nc (number of consecutive pairs of zeros or ones). Eventhough there are multiple ones or zeros in the TSR, they will not make an impact on the update value of wake-up interval until there are consecutive pairs of zeros or ones. Multiple consecutive zeros employ that the next wake-up interval should be increased in comparison to the previous value, whereas multiple consecutive ones imply that the next wake-up interval should be decreased.

First of all, it is important to make sure that the algorithm always converges towards a steady state and, secondly, the protocol needs to be extended for multi-hop environment. As the convergence of the algorithm is of primary concern, in this chapter a thorough analysis is conducted with an objective to understand the convergence behavior of TAD-MAC. Since the protocol is based on an adaptive algorithm, a rigorous testing is performed in a network simulator to understand the convergence behavior and other characteristics of the TAD-MAC protocol.

4.5 Heuristic-Based Approach for System Characterization

To exploit TAD-MAC protocol for general purpose WSN, it is necessary to show that the adaptive algorithm always converges towards a steady state. In this regard, variables of Eq. 4.1, I_{wu}, μ, and e are investigated through different methods and techniques (which are explained later in this section). The objective is to explore and understand the behavior of these variables to have a better understanding and more belief on TAD-MAC protocol. To observe and monitor the behavior of system variables and parameters, the system is characterized into open-loop and closed-loop forms with an aim to optimize the convergence time and reduce the processing by removing unnecessary variables (if possible) from Eq. 4.1. WSN system can not be modeled directly because of its non-linear behavior. Therefore, at first, it is important to model an open-loop behavior in order to understand the non-linearities that exist in the system. To achieve this objective a network simulator is used.

The system Eq. 4.1 contains two fixed parameters (initial wake-up interval I_{iwu} and reference time t_{ref}) and three variables (wake-up interval I_{wu}, update factor μ and error factor e). First, two variable μ and e are investigated against number of different (fixed) I_{wu} values (that is same as I_{iwu} value) and a fixed t_{ref} value in an open loop (without adaptation of the wake-up interval). Second, three variables (including I_{wu}) are explored in a closed loop where I_{wu} is updated at each wakeup and the experiments are conducted for number of different I_{iwu} values and fixed t_{ref}.

Let us first analyze the formal description of the adaptive algorithm to understand the complexity of the convergence problem. Let us note w_i, the vector of length L representing the content of the TSR register at time t_i such as

$$w_i = [w_{0,i}, w_{1,i}, \cdot \cdot \cdot \cdot w_{L-1,i}] \tag{4.5}$$

with

$w_i \in [0,1]$ and $\forall\ i \in [1, \cdot \cdot \cdot n_r]$, where n_r is the number of neighbor nodes. The number of ones and zeros and consecutive ones and zeros in the w_i are calculated through $N_{1,b}$, $N_{0,b}$ $Nc_{1,b}$ $Nc_{0,b}$.

$$N_{1,b} = \sum_{j=(b-1)L/2}^{b((L/2)-1)} w_j \tag{4.6}$$

$$N_{0,b} = \sum_{j=(b-1)L/2}^{b(L/2-1)} (1 - w_j) \tag{4.7}$$

$$Nc_{1,b} = \sum_{j=(b-1)L/2}^{b(L/2-1)} w_j \cdot w_{j+1} \tag{4.8}$$

$$Nc_{0,b} = \sum_{j=(b-1)L/2}^{b(L/2-1)} (1 - w_j) \cdot (1 - w_{j+1}) \tag{4.9}$$

where b $\in \{1, 2\}$. The algorithm converges to a steady state, i.e. $I_{wu}(t_{i+1}) = I_{wu}(t_i)$ as soon as the following condition is met:

$$Nc_{1,b}(t_i) = Nc_{0,b}(t_i) = 0 \tag{4.10}$$

$$Nc_{1,b}(t_{i+1}) = Nc_{0,b}(t_{i+1}) = 0 \tag{4.11}$$

Fig. 4.5 shows the model of the complete system from a sensor node perspective. The model is characterized through an error computation block which provides an input to the wake-up interval adaptation block to update the wake-up interval. Further, this value is used by a node to wake-up and sense the channel if there is any data (represented as $d(t_i)$) for it. It is important to note that wireless sensor networks has a non-linear behavior (due to variable traffic, clock drift, hardware/software latencies, channel variations, mobility etc.) and can not be modeled analytically to justify above conditions of convergence. Therefore, an experimental search is conducted (as explained later in this section) to show the behavior of all the variables and parameters of the adaptive algorithm.

The wake-up interval adaptation block (shown in Fig. 4.5) is basically an update part of the adaptive system shown in Fig. 4.4 and the error computation block is consists of error computation techniques such as weighting average algorithm, correlation between TSR and a reference TSR pattern. To realize the system characterization since WSN can not be modeled analytically due to non-linearities, a network simulator *WSNet* [27] [28] of version 9.07 is used.

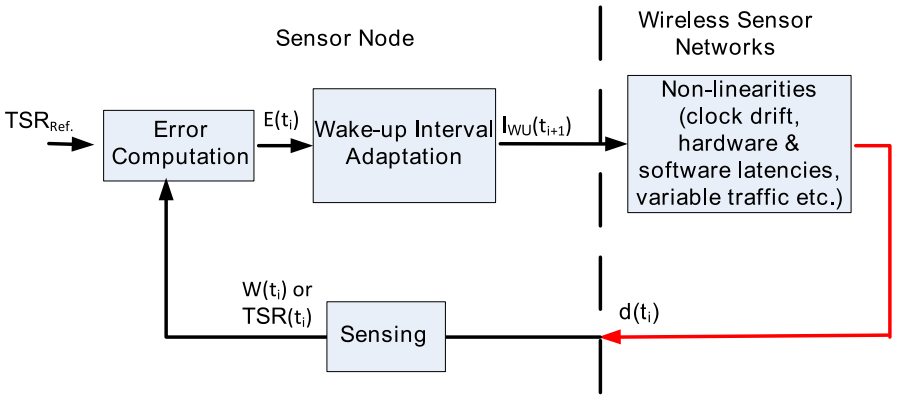


FIGURE 4.5: System model used to characterize the adaptive wake-up interval algorithm.

4.5.1 Experimental Setup of WSNet

The simulation setup consider different traffic rates, packet length of 16 bytes, free space propagation and BPSK modulation scheme. 0dBm of transmit power is used and receiver sensitivity of −92dBm is considered. The application model is basically controlled by the transmission rate of 1 *packet/sec* for open-loop experiments and 2 *packet/sec* for closed-loop experiments, whereas, in performance analysis (Sec. 4.6.1), both transmission rates are used in presented results. The network is simplified up to three nodes (two receive nodes and one transmit node) as the purpose is to understand the behavior of the variables and parameters of the protocol. At the MAC layer, maximum waiting time at the transmit node (before it can receive a beacon to transmit data) is set as $500ms$ and maximum waiting time for the acknowledgment at the transmit node is fixed as $5ms$. Whereas, maximum waiting time for receiving the data (at the receive node) after it transmits a beacon is $5ms$. A state machine is implemented at the MAC layer that includes an adaptive algorithm and all the necessary communication states of the TAD-MAC protocol. More details about the execution of state machine can be found in Sec. 4.6.

4.5.2 Open Loop System

The open loop form of the system, shown in Fig. 4.6, is divided into non-adaptive and real-time systems. In an open-loop form, as the wake-up interval is not updated and remains fixed therefore, it is termed as non-adaptive system. This fixed wake-up interval is used by a node to sense the environment (i.e. WSN) which is the second half and

termed as real time system. It is to mention here that, the wake-up interval is fixed with an initial value and it remains the same during the whole simulation.

A sensor node wakes up (with this fixed wake-up interval) to sense the environment (network) and receives a value (that can be either '1' or '0' as explained earlier in Sec. 4.4.1) which updates the contents of the TSR register. The contents of the TSR-register at time t_i are used to compute the controlled variables $\mu(t_i)$, and $e(t_i)$ and their behaviors are individually explained below. $\psi(t_i)$ is the product of the output of $t_{ref.}$ and the controlled variable. These controlled variables can be selected either as a combination of one or more variables or as only one variable to constitute ψ (depending upon the trade-off between performance and complexity). In the open-loop form even though $\psi(t_i)$ is computed, though as the wake-up interval is kept fixed it does not have any influence for sampling the next value.

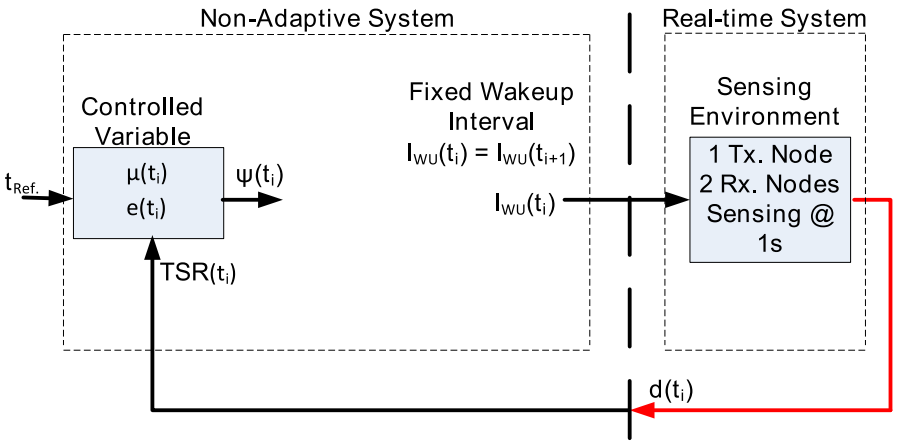


FIGURE 4.6: The open loop form of the system consists of two parts. The first one is a non-adaptive system that compute ψ (for the purpose of analyzing controlled variables) but does not used it to update the wake-up interval whereas, the second part is real-time system which is the wireless sensor network.

The purpose of the open loop form of the system is to evaluate the behavior of controlled variables under static conditions (fixed wake-up interval). In this regard, the simulation set-up consists of two receive and one transmit nodes, each transmit node has a transmission rate of 1 $packet/s$ which means that the coordinating (receive) node will have $500ms$ as the optimal wake-up interval to match the reference pattern. Number of different wake-up interval values are chosen starting from $100ms$ until $1000ms$ with an increment of $50ms$ and three variables are observed. It is to note that average values of all the variables are calculated over the length of the TSR-register, i.e. L (which is kept to a fixed value and is equal to 8).

4.5.2.1 Behavior of μ in open-loop

Variable μ is computed through Eq. 4.2 which is further based on Eq. 4.3 and Eq. 4.4. Fig. 4.7 shows the behavior of μ under different wake-up interval values. It can be observed as the wake-up interval reached $500ms$ then the μ value becomes 0, whereas on the two other extremes (such as 100ms and 1s) it has maximum opposite value. This is due to the fact that with $500ms$ value of wake-up interval the contents of TSR-register perfectly matches with the desired pattern, which means that the number of zeros and ones are equal with no consecutive zeros or ones in TSR and consequently μ is zero. Each fixed wake-up interval value results in fixed periodic sequence (TSR-Register contents) and an average value is calculated from the periodic sequence which is shown in Fig. 4.7. It can be seen that the variation of μ is monotonic and almost linear, hence μ variable always helps to converge the algorithm towards a steady state as the second order derivative of this monotonic function is zero.

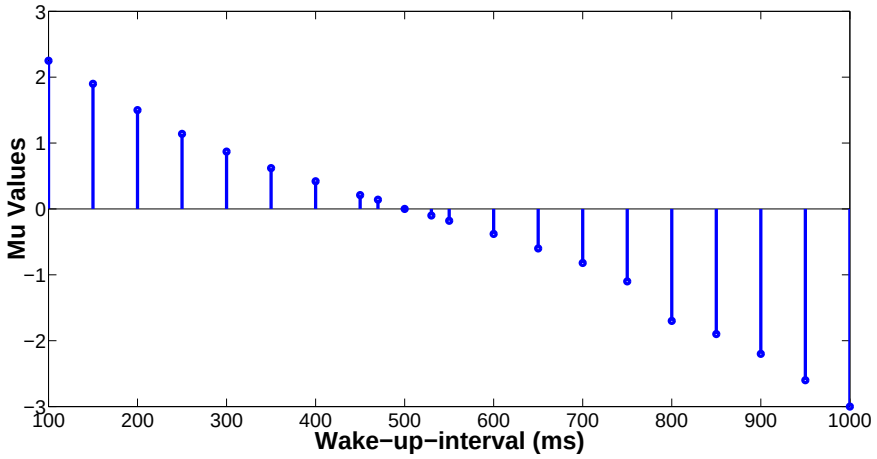


FIGURE 4.7: Variations of mu variable against different (fixed) wake-up interval values.

4.5.2.2 Behavior of e in open-loop

Variable e is explored through two different techniques, first by using correlation and second by counting the number of ones in TSR. Fig. 4.8 shows the behavior of variable e computed through correlation. The correlation is performed between a reference pattern and the contents of TSR. Mean and absolute values of correlation are calculated over the length of TSR (i.e. for one current and seven previous values of correlation) and are shown in Fig. 4.8. It can be seen that at a $500ms$ value of wake-up interval, both sequences (i.e. reference and TSR) perfectly match therefore, the mean value of

correlation is 0 whereas, the absolute of mean value is 8 (maximum). This is due to the fact that generally correlation can not be negative (which means it can only contribute to increase the wake-up interval when used for the adaptation of wake-up interval in TAD-MAC). Therefore, reference pattern with a sequence $[1\ -1\ \ 1\ -1\ \ 1\ -1\ \ 1\ -1]$ is used, and TSR contents are filled with -1 instead of 0 if no data is received. It can be observed that correlation values in two different forms have predictable behavior (at 500ms) as explained above and can be seen in Fig. 4.8. However, it has non-monotonic behavior for most of the wake-up interval values.

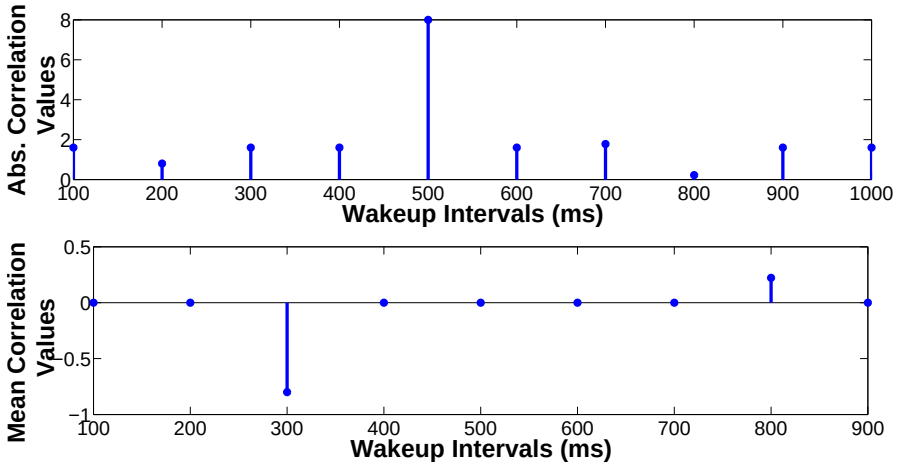


FIGURE 4.8: Behavior of e variable against different (fixed) wake-up interval values calculated through correlation technique. First is the mean of the absolute value of correlation and second is only the mean of correlation.

Fig. 4.9 shows the variations of e computed by counting the number of ones in TSR against the number of different fixed wake-up interval values. It can be seen that e varies linearly with regards to wake-up interval, moreover it is 0 at perfect matched wake-up interval value i.e. $500ms$. It is important to mention here that even though e (computed by counting number of ones) seems a perfect candidate for the computation of wake-up interval update (for the closed-loop system explained in Sec. 4.5.3), but it can result in miss-match with desired sequence under 60 different combinations out of 256 (for TSR length equals to 8), as the number of ones and zeros in these sequences are equal. These patterns can be calculated by subtracting all the combinations (except when number of zeros and ones are equal i.e. 4 for a length of 8) from the total possible combinations and can be expressed as, $2^L - C_0^L - C_1^L - C_2^L - C_3^L - C_5^L - C_6^L - C_7^L$. Therefore, e computed by counting only the number of ones is not a precise technique as it will be zero in many undesired sequence of convergence.

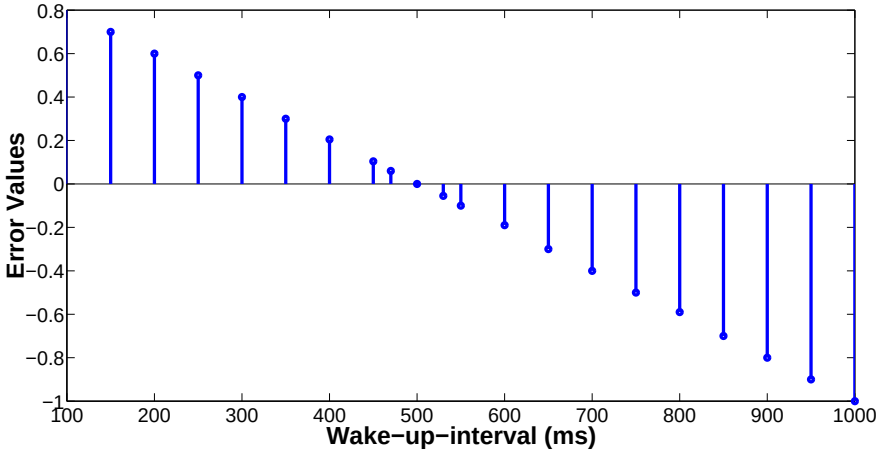


FIGURE 4.9: Variations of e variable against different (fixed) wake-up interval values computed by counting number of ones in TSR.

To conclude, e calculated through a correlation technique does not has linear or monotonic behavior and it can not be considered for closed-loop system. Whereas, e calculated by counting the number of ones has a linear behavior but there are some specific patterns in which its computation is not accurate. Though, in next Sec. 4.5.3 it has been used in combination with μ to observe if it helps to achieve the fast convergence.

To conclude on the open-loop system, it is observed after characterizing the controlled variables that μ and e (computed by counting number of ones) are suitable techniques to be considered in closed-loop form of the system. Whereas, variables e (computed through correlation technique) have non-linear behavior and can not be considered for adaptation of wake-up interval in the context of real network.

4.5.3 Closed-Loop System

The closed-loop form of the system, shown in Fig. 4.10, is divided into adaptive and real-time systems. The main difference between open-loop and closed-loop forms is that ψ is used to update the next wake-up interval that is used by a sensor node to wake-up and sense the environment (i.e. WSN) in closed-loop. At the start, the wake-up interval is initialized and afterwards it is updated at each time index (wakeup instant). A sensor node samples the environment with this updated time interval and receives a value $d(t_i)$ which updates the contents of the TSR register.

The content of the TSR-register is used to compute the controlled variables μ and e (which are selected based on the monotonic and linear behavior respectively). In the

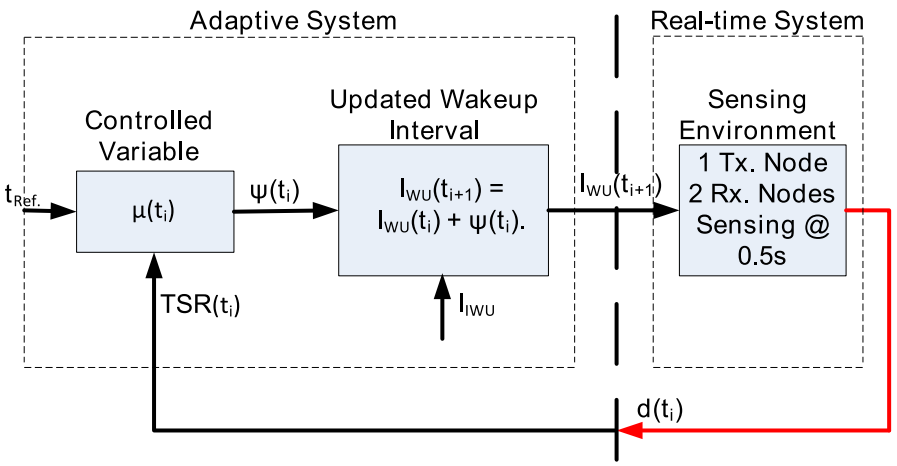


FIGURE 4.10: Closed-loop form of the system considers the influence of feedback into the computation of the update of the wake-up interval in the adaptive system. It is to note that only μ and e are considered as controlled variables in the adaptive system.

closed-loop form, ψ can be calculated as a combination of μ and e or only through μ depending upon which of these two options provides best convergence speed and at the same time with minimum processing power. For both cases, the experiments are conducted with fixed data transmission time of $500ms$ between two packets (packet rate) and receive node adapts its wake-up interval according to this rate. In this regard, different initial wake-up interval values are used ranging from $100ms$ (that is much less than packet rate) up to $1000ms$ (which is much more than packet rate). It is to mention here that, as the receive node reaches a steady state, it has a value of wake-up interval exactly half to that of packet rate (which is the target rate of convergence), because of the desired sequence [10101010...].

First, the experiments are conducted by using only μ as a control variable.

4.5.3.1 Closed-loop form validation of μ

In order to achieve the best convergence time which helps to reduce energy consumption and delay with least processing power (i.e. by utilizing less parameters), following experiments compute ψ by using only μ.

Fig. 4.11 shows the behavior of wake-up interval adaptation for number of different initial wakeup starting values. It can be seen that by using only μ as a controlled variable for the update of wake-up interval helps to improve the convergence speed. Moreover, the adaptation resembles to typical adaptive or control systems with monotonically decaying

towards a steady state without any long oscillations. Fig. 4.11 is divided into zoomed version and full version in which traffic rate also changes from $500ms$ to $1000ms$ and further $400ms$ and it is observed that under variable traffic receive node adapts very fast to new steady state value.

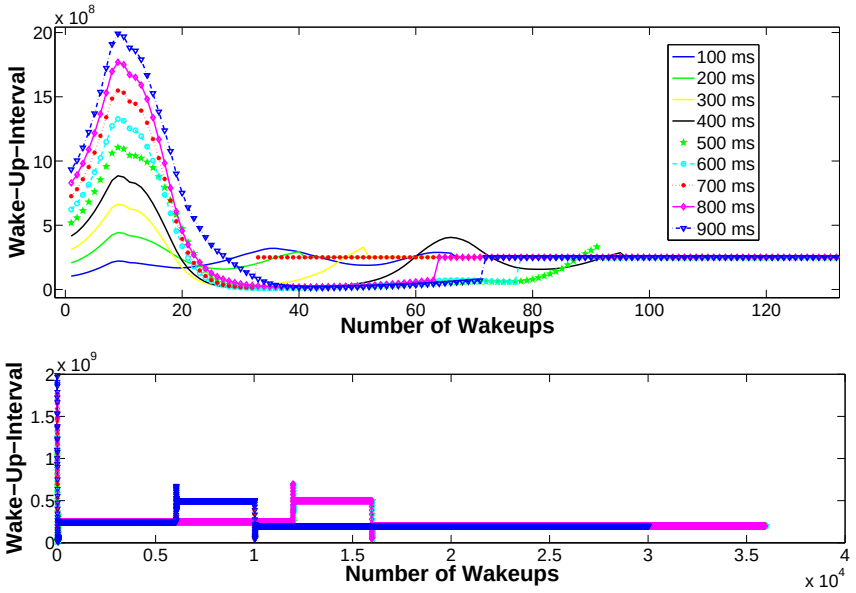


FIGURE 4.11: Wake-up interval adaptation for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one, also showing the behavior of convergence under variable traffic.

Fig. 4.12 shows the variations of μ in closed loop form with adaptation of wake-up interval. It can be seen that μ variations in all different initial wake-up interval values is a decaying function and reaches back efficiently towards a steady state as soon as the traffic rate changes.

Further, we will explore if we can improve the performance by using also e variable.

4.5.3.2 Behavior of the combined μ and e in closed-loop

In order to achieve the best convergence time that can help to reduce energy consumption and delay, μ and e are used in combination to compute ψ in following experiments.

Fig. 4.13 shows that the receive node reaches a steady state with a wake-up interval value of $250ms$ for all the initial values of wake-up interval. It can be seen that for initial values starting with $200ms$, $300ms$ and $400ms$ receive node takes longer time to converge. It has been observed that with these initial values the receive node reaches such wake-up interval values that it starts oscillating with nearly periodic pattern (with very

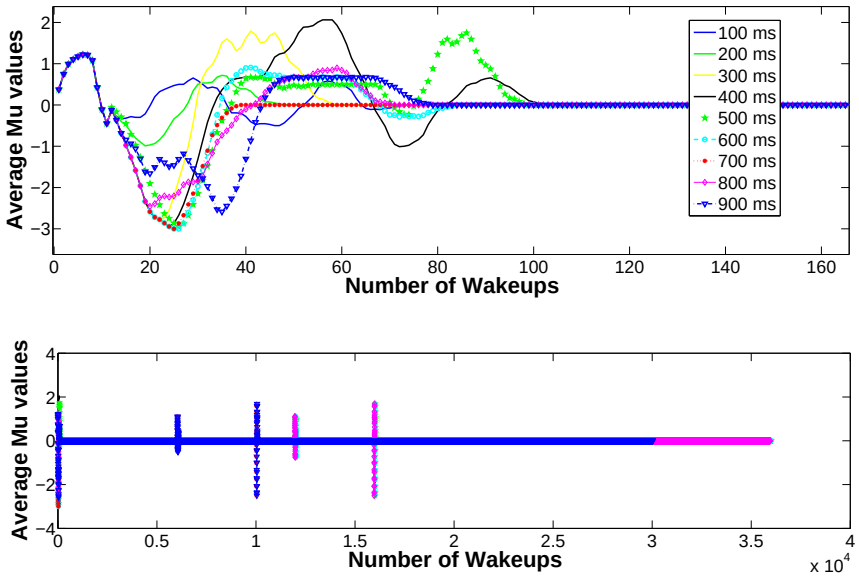


FIGURE 4.12: Behavior of *mu* for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one.

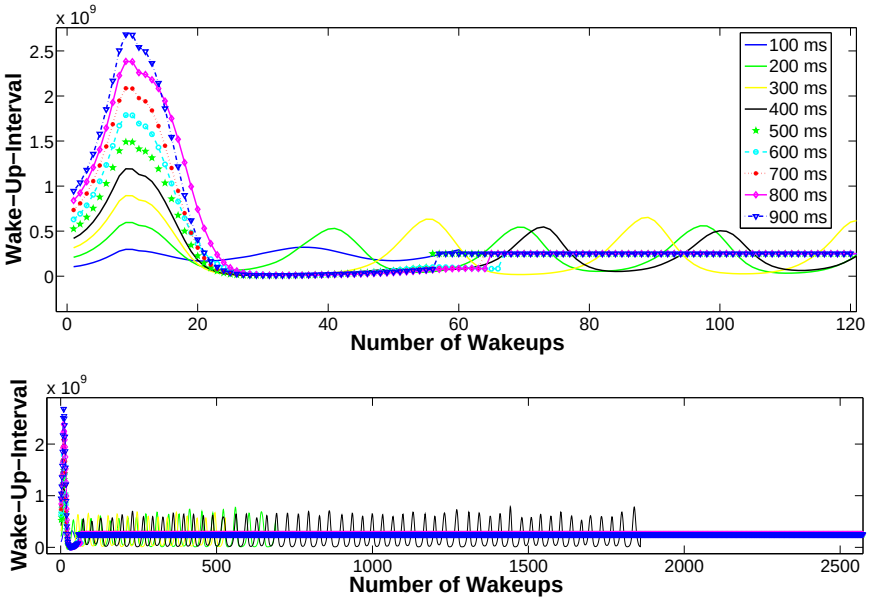


FIGURE 4.13: Behavior of wake-up interval adaptation for number of different initial wake-up intervals. Few initial wake-up interval values result in oscillation and hence converge much later than others.The top figure is a zoomed version of the bottom one.

little change) in terms of TSR contents and consequently the controlled variables ($\mu + e$) takes longer time to converge as can be seen from Fig. 4.14 and 4.15. Moreover, due to the process of locking of wake-up interval and wake-up time (which is explained in detail in Sec. 4.6), the oscillations abruptly converge towards a steady state. It is important to mention here that both μ and e can reach to 0 value which means steady state even if TSR contents does not perfectly matched with the desired sequence of convergence as explained in Sec. 4.5.2.2, hence resulting in false convergence. Therefore, two processes (i.e. tracking and locking) are used to make sure that the accurate steady state value can be obtained. These two process are explained in detail in Sec. 4.6.

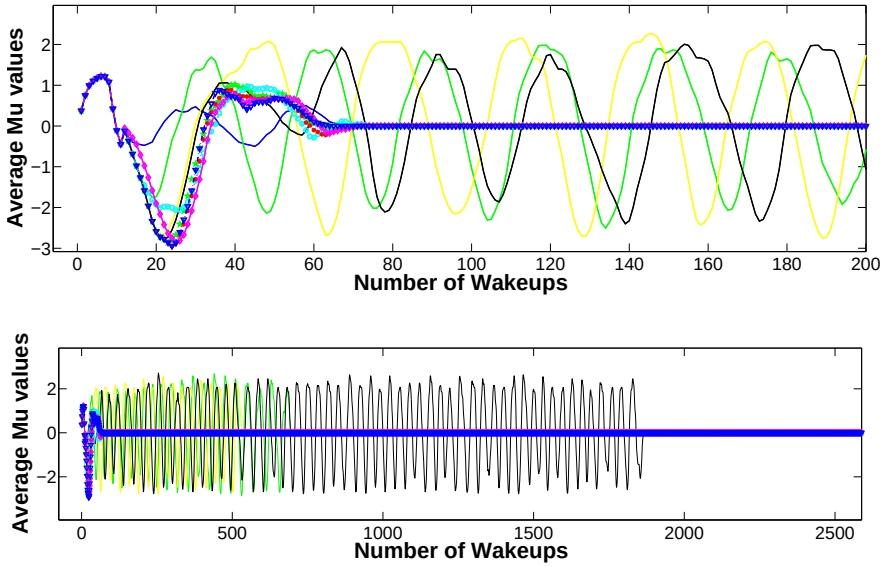


FIGURE 4.14: Variable *mu* behavior for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one.

To conclude, combined effect of μ and e variables for adaptation of wake-up interval shows no improvement in the convergence speed. As e counts only the number of ones and its impact already exists in the computation of μ.

4.5.3.3 A brief conclusion on closed-loop system

The closed-loop form using only μ performs much better both in terms of convergence speed and with minimum processing power along with an adaptive system without oscillations. Number of different sensing rates are verified and few of them are presented with traffic statistics and their results in next section. It is important to mention here that once a node converged towards a steady state it takes much less time to converge

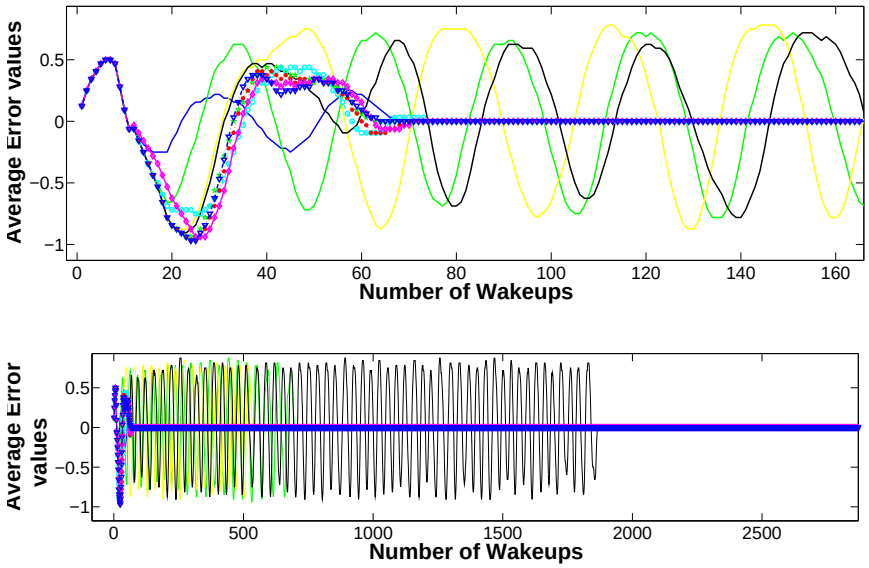


FIGURE 4.15: Behavior of variable e for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one.

back after the traffic rate changes, which shows that TAD-MAC performs much better in comparison with other protocols under variable traffic as can be seen in [15].

4.6 Implementation And Performance Analysis

In this section the implementation details of TAD-MAC protocol in a network simulator are explained. Moreover, performance metrics such as energy consumption, latency, delivery ratio and convergence time are presented.

Applying an adaptive algorithm using μ to reach a steady state is not sufficient because μ can be zero (which means no update in the wake-up interval) through 10 different sequences whereas only one of them is a desired sequence of convergence. To resolve this issue two different processes are implemented, i.e. tracking and locking. During the tracking phase the algorithm allows the node to converge to a steady state (that can be any of the 10 sequences) which can result in μ equals to zero. However, the wake-up interval achieved with these sequence may not be the best value. Therefore to ensure that the node converges only to [10101010] sequence, wake-up interval is locked only when a node has converged to it and not when it has converged to other sequences. Locking process is divided into two steps, in the first step the difference of the wakeup

time between the transmit and receive node is calculated and added in the current wake-up interval (at receive node) to compute next wake-up interval. In the second step the receive node adjusts its wakeup time according to transmit node by knowing the waiting time of the transmit node (that is transmitted in the data packet i.e. MAC frame header).

Fig. 4.16 shows the flow of the state machine used to simulate TAD-MAC protocols in the WSNet simulator. The flow depends upon the existence of data packets; for example, if a node has a data packet to transmit it follows the transmit mode (*Tx Flow*) and if it has no data it follows the receive mode (*Rx Flow*). The scheduler moves to the next state based on the occurrence of an event, otherwise it keeps on waiting until the timer expires and then it proceeds backward. For example, a node wants to transmit a data packet and it waits for the wakeup beacon from the receive node; if it does not receive a beacon in the *maximum waiting time for beacon* then it goes to 'sleep state'. On the other hand if a transmit node receives a beacon from the receive node it moves forward to next state, i.e. the clear channel assessment (CCA) state.

The TSR contents are updated in the *RX Flow* during the 'wait for data state'. If a node receives one data, $'1'$ is inserted in TSR otherwise $'0'$ is inserted. Whereas the adaptive algorithm is implemented in the 'adaptive algorithm state' just before going to sleep. This state computes the variable μ, the *next wake-up interval*, and the *next wakeup time* of the receive node. Executing 'adaptive algorithm state' just before going to sleep has two advantages, first it avoids any delay during the communication between two nodes and second radio transceivers can be switched off after the completion of transmission and reception, which avoids extra cost (energy, delay) due to processing of an algorithm as only the microprocessor remains active.

The 'adaptive algorithm state' also evaluates which specific neighbor (transmit node) to be served on the next wakeup with a wakeup beacon by calculating the most nearest time. For this purpose every (receive) node contains a bank of TSR as shown previously in Fig. 4.3 where each TSR (that is specific to each neighbor) evaluates the next wakeup time for that neighbor. During rigorous simulations in WSNet it is found that a conflict (due to overlapping) of serving a wake-up beacon among multiple nodes can occur before all the nodes reached a steady state. In order to deal with such conflicts, certain priorities are defined at the receive (coordinating) node.

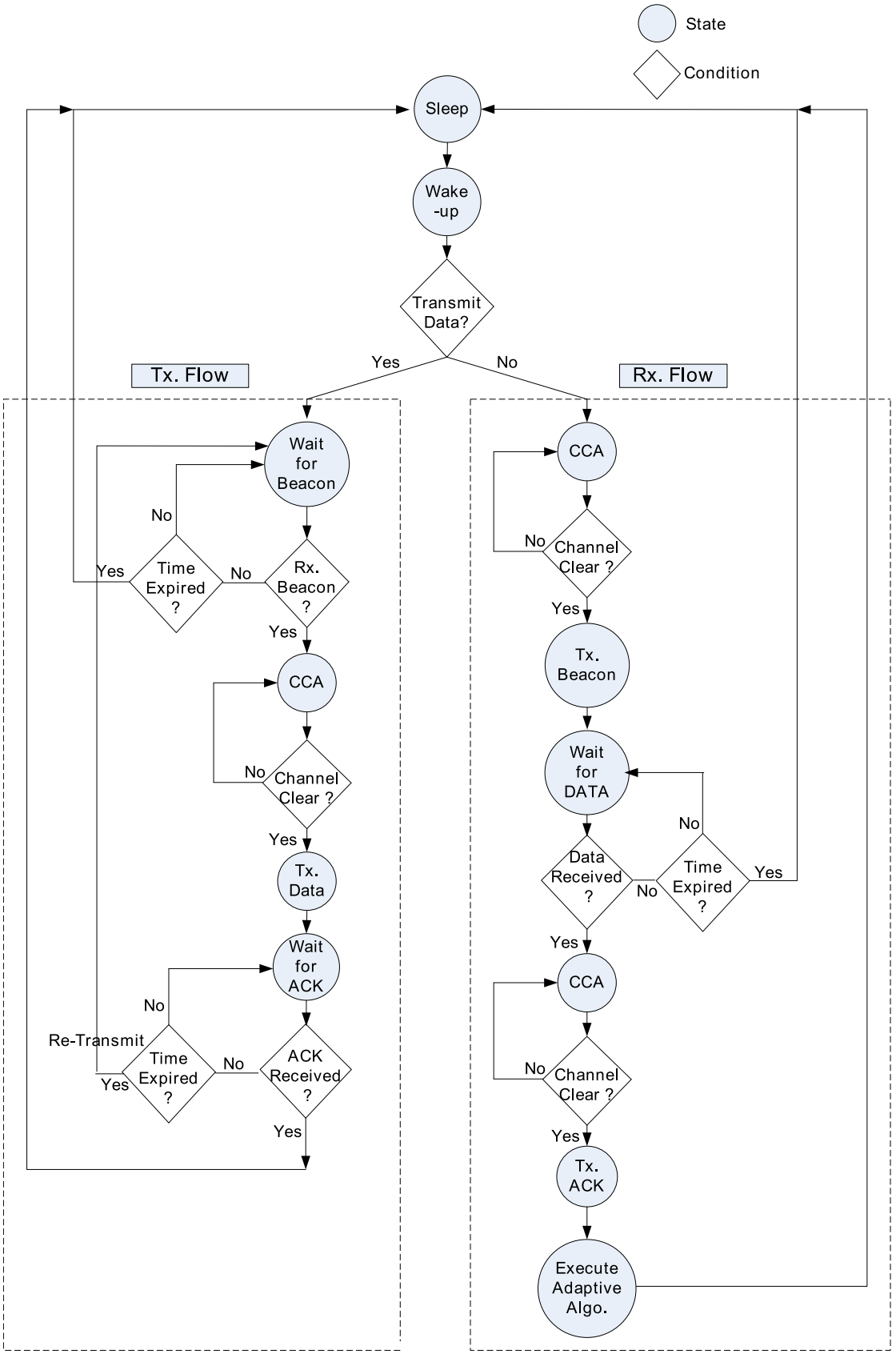


FIGURE 4.16: State machine being used to simulate TAD-MAC protocols in WSNet simulator. The scheduler moves to the next state based on the occurrence of an event, otherwise it keeps on waiting until the timer expires and then it proceeds backward.

- **Strict Priority**: If a (transmit) node 'A' is already synchronized, whereas the (transmit) node 'B' is not yet synchronized with regards to receive node 'R', then node 'B' has a higher priority over node 'A', when a wake-up beacon conflict occurs between them. The node 'B' has higher priority because of two reasons. First, node 'A' which is already converged to a steady state with respect to 'R' demands only minor update at 'R' in its corresponding TSR. The receive node will insert dummy information $'1'$ if it did not received data packet when it last time served a wake-up beacon to node 'A', whereas, $'0'$ if it had received data last time) to keep the convergence of node 'A' intact. Second, by giving higher priority to the node 'B' which is not yet synchronized allows its evolution/convergence process to not deviate as it may be close to converge. It is important to mention here that, if the data rate changes at the time of conflict, then multiple ones or zeros will appear in TSR (within 2 or 3 wakeups) and as a result the process of re-convergence to another traffic rate will be started.

- **Lenient Priority**: Either if node 'A' and node 'B' are both already synchronized with regards to 'R', or both are not yet synchronized, then any one of them can be selected to be served with a wakeup beacon. Though, it is important to note that in case of both 'A' and 'B' are synchronized and suppose node 'A' is selected, then the TSR contents of node 'B' must be updated depending upon if it has to receive '1' or '0', as explained above.

Moreover, other than above mentioned conflicts, there are moments when the wakeup times of the transmit nodes are not exactly overlapping but they are very close. In such circumstances, a node that has been served with the beacon first is still in the process of communication with receive node whereas, during this period the wakeup time to serve another node has passed away. To deal with this issue, above priorities are utilized in a sense that the TSR contents at the receive node are updated (with either '1' or '0') of the second node (whose time has passed away) at an expense of loosing one data packet from the second node if it has to transmit a data packet on that moment. Of course the inclusion of a FIFO (of few packets size at the transmitter) can avoid this packet loss. Moreover, it is important to mention here that as this specific case occurs the converged sequence [10101010] may become [11010101] and can unlock the convergence. Therefore, such specific cases are dealt with special care by identifying and counting the *number of miss* that node contains and accordingly by inserting '0'

in between '1' to remain locked. Furthermore, special distinction/care has been taken for varying traffic and more specifically for increasing traffic where the transmit node also transmits multiple data, in that case as multiple ones exceeded the *number of miss* and consequently transmit node will no more remained converged as the traffic rate has changed.

4.6.1 Performance Analysis

Life time analysis is estimated and presented in next chapter for different preamble sampling MAC protocols and for different radio chips, which shows a clear gain when using TAD-MAC protocol for WBASN. The focus here is to evaluate performance metrics of the evolution phase for reaching to a steady state because as the nodes are converged then all the performance metrics are optimized. In this regard many experiments are conducted with varying traffic rates and two of them are shown as an examples.

Different performance metrics have been evaluated to show the results of TAD-MAC protocol, these include energy consumption, latency, convergence time and delivery ratio. In this regard, number of data and control packets transmitted and received for reaching a steady state are exploited from the network simulator whereas our precise real-time hybrid energy model presented in Chapter 3 is used for the evaluation of accurate energy consumption.

The experimental setting are the same as explained in Sec. 4.5.1. The results presented in closed-loop system by using only μ are used here in performance analysis as it represents one of the data rate (i.e. 500ms). Whereas, the results for another data rate (i.e. 1s) are further extracted and are presented.

TABLE 4.1: The number of beacons and acknowledgments transmitted by the receive node and number of data packets received by the receive node before reaching the convergence for sensing rate of 1s. Three cases (average, best and worst) are presented based on the performance of three (average, best and worst) initial wake-up interval values.

	Best Case	**Average Case**	**Worst Case**
Number of Beacons Transmitted	22	72.9	150
Number of Data Packet Received	8	37.6	73
Number of Ack. Transmitted	8	37.6	73

TABLE 4.2: The number of beacons and acknowledgments transmitted by the receive node and number of data packets received by the receive node before reaching the convergence for sensing rate of 500ms. Three cases (average, best and worst) are presented based on the performance of three (average, best and worst) initial wake-up interval values.

	Best Case	**Average Case**	**Worst Case**
Number of Beacons Transmitted	33	53	83
Number of Data Packet Received	16	31	42
Number of Ack. Transmitted	16	31	42

Tab. 4.1 and Tab. 4.2 shows the number of beacons and acknowledgments transmitted by the receive node and number of data packets received by the receive node before reaching a steady state. Tab. 4.1 represents the results for sensing rate of $1s$ whereas, Tab. 4.2 represents the results for sensing rate of $500ms$. Generally traffic loads differ from node to node depending upon their position in the networks, sink node and the ones close to it have much more load in comparison to the others. The purpose of Tab. 4.1 and Tab. 4.2 is to understand how many data and control packets are necessary before the transmit and receive nodes are become synchronized.

Tab. 4.1 and Tab. 4.2 contains three different cases; *best case* is the fastest convergence case, *worst case* takes the longest time to converge and then the *average case*. These three cases corresponds to different initial wake-up interval values (ranging from 100ms to 1s) that are used to extract these characteristics. It is observed that by using $100ms$ as an initial wake-up interval value nodes always converge faster for both sensing rates. Moreover it can be noticed that there exists a great variation between Tab. 4.1 and Tab. 4.2 and then, from best to worst cases of two traffic rates. Whereas, the average case is much closer in both traffic rates.

TABLE 4.3: Performance metrics for sensing rates of 1s before reaching a steady state.

	Best Case	**Average Case**	**Worst Case**
Number of Packet Lost	2	12.5	30
Latency (sec.)	1.52	6.57	15.33
Convergence Time (sec.)	8.43	45.5	72.59

TABLE 4.4: Performance metrics for sensing rates of 500ms before reaching a steady state.

	Best Case	**Average Case**	**Worst Case**
Number of Packets Lost	0	8	12
Latency (s.)	2.18	4.34	5.7
Convergence Time (s.)	9	21.8	38

Tab. 4.3 and Tab. 4.4 shows the performance metrics results for three different cases, i.e. *best case*, *average case* and the *worst case* that corresponds to different initial wake-up interval values for two traffic rates. Tab. 4.3 represents the results for sensing rate of $1s$ whereas, Tab. 4.4 represents the results for sensing rate of $500ms$. These performance metrics are number of packets being lost, latency (i.e. idle listening) and convergence time before reaching a steady state and are explained below.

- **Packet Losts**: It is the total number of data packets being lost before reaching towards a steady state. Since we did not consider any FIFO in our implementation, a packet is considered to be lost if the waiting time to receive a beacon expires. It can be seen from Tab. 4.3 and Tab. 4.4 that, on one extreme number of packets being lost can be 0 which means 100% delivery ratio, whereas, on the other extreme number of packets being lost can reduce the delivery ratio until 70% and hence increase the time of convergence and as a result impact on the energy consumption as well.

- **Latency**: It is defined as the time a transmit node has to wait before it transmits a data packet. In TAD-MAC protocol we termed this time as *idle listening*, further the total time (which is an addition of all the *idle listening* until a node reaches a steady state) is shown in Tab 4.3 and Tab. 4.4. Since it is the transmit node who has to wait for the beacon before being able to transmit therefore all the contribution and impact of latency is from transmit node. For the best case it is 1 or 2 seconds, whereas in the wort case it can reach up to $15s$.

- **Convergence Time**: It is the total time before a transmit node converged with respect to receive node. The fast convergence speed is the best in terms of energy,

latency and delivery ratio. The best and worst convergence times are $9s$ and $73s$ respectively whereas on average it is around half a minute.

The results for the energy consumption of the receive and transmit nodes are presented in Tab. 4.5 and Tab. 4.6 respectively. The detail computation are explained below.

- **Energy Consumption (RX)**: It is the total energy consumed by a receive node before reaching a steady state. Tab. 4.5 shows the energy consumption at receive node by calculating the total number of beacons being transmitted, total number of data packets being received and total number of acknowledgments being transmitted. The best, worst and average cases corresponds to different initial wake-up interval values. Tab. 4.1 and Tab. 4.2 are used and the energy consumption of the receive node before reaching a steady state is calculated through accurate energy model presented in Chapter 3. Tab. 3.2 provides the energy consumption of wake-up beacon, acknowledgment, and data transmission/reception, that are used to calculate the energy consumption presented in Tab. 4.5.
- **Energy Consumption (TX)**: It is the total energy consumed by a transmit node before reaching a steady state. Tab. 4.6, shows the energy consumption of the transmit node due do transmission of data packets as well as due to waiting time (i.e. *idle listening*) to receive wakeup beacon and are presented similarly for two different data rates (i.e. 1s and 500ms). The total number of data packets transmitted by the transmit node is the sum of number of packets received (shown in Tab. 4.1 and Tab. 4.2) and the number of packets being lost (shown in Tab. 4.3 and Tab. 4.4). Whereas, the *idle listening* is the latency that is presented in Tab. 4.3 and Tab. 4.4 for two data rates. For the computation of energy consumption values hybrid energy model is utilized.

4.6.1.1 A brief conclusion on performance analysis of TAD-MAC

To conclude, performance analysis of TAD-MAC protocol has shown that the protocol is very optimized in terms of energy consumption as well as latency. On average it takes between 30s to 40s to converge towards a steady state. Moreover, the adaptive algorithm converges efficiently towards various traffic rates and is validated through experiments.

TABLE 4.5: Total energy consumed by a receive node before reaching towards a steady state for sensing rate of 1s and 500ms. Tab. 4.1 corresponds to sensing rate of 1s and Tab. 4.2 corresponds to 500ms and are used to calculate the energy consumption for average, best and worst cases.

Rx Energy Consumption		**Best Case**	**Average Case**	**Worst Case**
Beacon (mJ)	Tab. 4.1	1.3	4.4	9.1
Data (mJ)	Tab. 4.1	1.2	5.5	10.8
ACK (mJ)	Tab. 4.1	0.04	1.9	3.7
Beacon (mJ)	Tab. 4.2	2.0	3.2	5.0
Data (mJ)	Tab. 4.2	2.4	4.6	6.2
ACK (mJ)	Tab. 4.2	0.08	1.6	2.1

TABLE 4.6: Total energy consumed by a transmit node before reaching towards a steady state for transmission rate of 1 *packet*/s and 2 *packet*/s. Tab. 4.3 and Tab. 4.1 is used for evaluating the energy consumption of idle listening and data transmission for 1 *packet*/s. Tab. 4.4 and Tab. 4.2 are used for 2 *packet*/s.

Tx Energy Consumption		**Best Case**	**Average Case**	**Worst Case**
Idle Listening (mJ)	Tab. 4.3	100	450	1005
Data Transmission (mJ)	Tab. 4.1	0.5	2.6	5.0
Idle Listening (mJ)	Tab. 4.4	150	297	390
Data Transmission (mJ)	Tab. 4.2	1.1	2.1	2.9

4.7 Conclusion

The idle energy consumption is the dominant energy waste in WSN and is few multiples of the energy used in the actual transmission. In this chapter we proposed a novel MAC protocol (TAD-MAC) that allows the sensor nodes to adapt themselves dynamically according to the traffic. The dynamic adaptation of TAD-MAC results in ultra low energy consumption from idle listening, overhearing, collisions and unnecessary wake-up beacon transmission. Each node has a traffic status register bank which contains the traffic status according to the data received from all the neighbor nodes. Each sensor node adapts its wake-up interval through TSR in such a way that idle listening is minimized and the convergence to a steady state is achieved through continues update of wake-up interval. In order to make sure that TAD-MAC always converges towards a steady state value an exhaustive search method is used to characterize the system variable μ that is used in the computation of proposed adaptive algorithm. Number of different initial wake-up interval values are used to observe the behavior of μ in open loop and closed loop forms that are used to characterize the system. It is shown that μ has a monotonic behavior in open loop

form, which means the second order derivative of this variable is zero, hence the algorithm always converges. Whereas, in closed loop form it varies like a decaying function that reaches zero within short time and consequently it shows that the algorithm reaches a steady state. TAD-MAC protocol is implemented in a network simulator (WSNet) and various performance metrics such as energy consumption, latency, convergence time and delivery ratio are defined and evaluated through rigorous simulations. In this regard, traffic statistics are utilized from simulator to estimate performance metrics. Best, worst and average convergence time results are presented that are $8s$, $72s$ and $35s$ respectively, whereas algorithm also converges very fast (within $8s$) whenever the data rate changes due to the traffic variations. It is also shown that if no FIFO is used to buffer data packets the delivery ratio can be reduced up to 70% before a node reaches a convergence state.

Chapter 5

Application of TAD-MAC Protocol for Wireless Body Area Sensor Networks

Traffic-aware dynamic MAC (TAD-MAC) proposed in the previous chapter helps significantly to optimize the energy consumption for wireless sensor networks. It has been shown in TAD-MAC that the coordinating node converges efficiently towards a steady state for different traffic rates of various neighbor nodes. In this chapter, we apply TAD-MAC to wireless body area sensor networks (WBASN) to effectively utilize the energy and latency optimizations proposed in TAD-MAC. First we present the context and application scenarios of WBASN followed by related work. Then the proof-of-concept is presented by mapping TAD-MAC into WBASN configurations and explaining simulation results. Finally the performance evaluation is presented by comparing different preamble sampling MAC protocols under fixed and varying traffic of WBASN and utilizing different radio transceiver chips.

5.1 Introduction to WBASN Context

A wireless body area sensor network (WBASN) is a special purpose wireless sensor network (WSN) that can be incorporated within different networks such as a wireless local area network (WLAN) to enable remote monitoring for various environments [128].

One of the applications of WBASN is health care monitoring, where a number of patients can be observed, diagnosed, prescribed remotely. WBASN can be classified as invasive (in-body) or non-invasive (on-body) networks [115]. Smart sensor nodes can be connected to various parts of the body or fabricated inside clothes to transmit the information to the base station [127]. Thus, WBASN has emerged as a promising alternative to traditional wired network systems for medical environment with significant impact on the rehabilitation and improved quality of life of the patients along with minimum personal monitoring [122].

WBASN can be applied to different scenarios such as monitoring of an individual patient from home, monitoring of few patients from intensive care units (ICU), and monitoring of many patients from hospital wards. Typically few sensors (less than 10) are connected on the body to monitor the physiological data emanating from the body as shown in Fig. 5.1. Generally in these scenarios there is no requirement for large network; single-hop communications with star or clustered-based topologies are most popular in WBASN [93], [120]. Moreover, a coordinating node often controls all the communications between the nodes as well as with access points. In this context, energy consumption and latency are two important design constraints along with the miniature size of the sensor nodes. The nodes that are implanted inside the body require high energy efficiency, in other words the lifetime of the sensor nodes and network needs to be prolonged by the use of ultra-low power technologies and protocols [127], [32].

Typically in WBASN, physiological data of various parts of the body are transmitted over the network. Different nodes connected inside or on the body have great variation in terms of data rate. For example, in an emergency room for heart patients, continuous monitoring of electrocardiogram (ECG), oxygen, body temperature and blood pressure could be required. In such a case, the sensors connected for these purposes have huge variations in their data transmission rates as presented in [9]. Further, to emphasize on the traffic variations, for example, there are 10-to-20 times more packets per second from *ECG* monitoring in comparison to *pulse rate* or *body temperature* monitoring [9]. For example, the data rate of *EEG*, *ECG* and *EMG* according to [46], [53], and [18] ranges from 32 to 256 to 600 kbps respectively. Due to these high data rates, these signals are normally compressed and even then there exists a big difference in data rate among compressed signals due to different compression algorithms. For that matter,

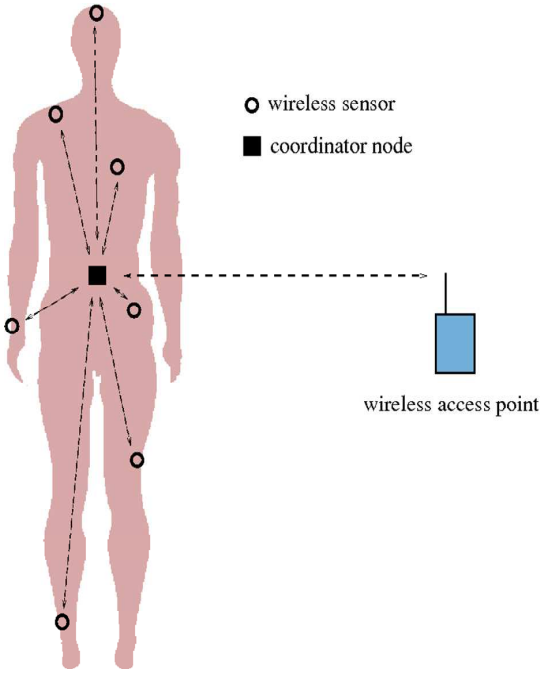


FIGURE 5.1: Typical body area sensor networks consist of few nodes connected on the body and a coordinating node to control the communication between the nodes as well as with access points.

[65] provides the *ECG-compressed* rate equal to 1.2 kbps, whereas [47] provides *ECG-compressed* rate equal to 3.96 kbps. Similarly, for the low data rate (such as temperature and heart rate) monitoring, there exist variations based on the condition of the patient being monitored [46], [18], [87], [53]. Therefore, to accommodate these variations of traffic, it is very important to utilize a dynamic MAC protocol that can adapt to various nodes according to their traffic requirements.

The network configuration is another important design attribute of WBASN. For invasive sensors, it is difficult to communicate with each other due to the complexity of the human body, therefore, a surgeon avoids in-body communication between sensors. Typically, a star topology is used for invasive WBASN, where all the in-body sensors are connected to a coordinating node which has more battery power and processing capabilities with regards to the implanted nodes. For the case of non-invasive sensors various options can be selected depending upon the inter-communication among the nodes, number of nodes and patients. For monitoring home-based or intensive care units where few nodes are on every patient body, star or mesh network is suitable and multiple coordinating nodes can even be configured. On the other hand, for hospital wards multiple coordinators or multi-hop based routing mechanisms are necessary. In this context a hybrid network topology, as illustrated by Fig. 5.2, seems to be the most suitable configuration for WBASN.

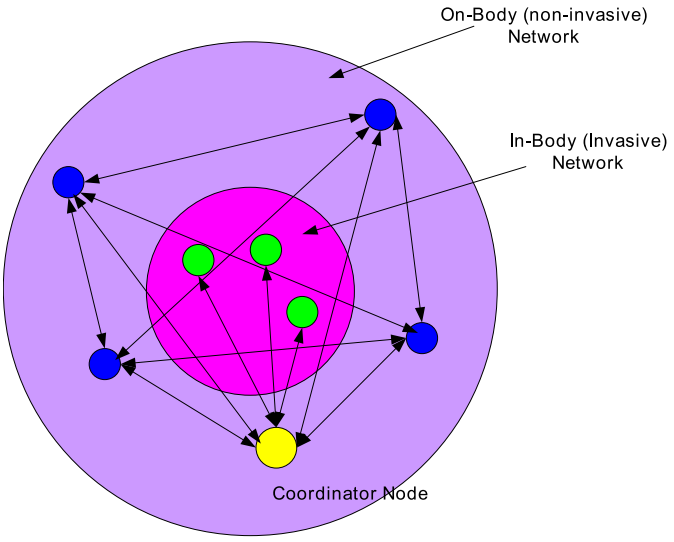


FIGURE 5.2: Hybrid network topology: invasive sensor nodes are connected with a coordinator through star network, whereas non-invasive nodes are configured through mesh topology.

It is necessary that the MAC layer incorporates in its design phase varying traffic (among sensor nodes) and variable traffic (within sensor node) along various dynamics of application scenarios and network configurations as explained above. In this regard, the TAD-MAC protocol presented in previous chapter is designed by keeping the importance of application dynamics and network configurations.

In this chapter TAD-MAC is applied to exploit maximum energy efficiency for WBASN. The presented protocol targets both invasive and non-invasive body area networks by considering a hybrid network topology which includes a star network for in-body and a mesh network for on-body WBASN. For an invasive network, as the implanted nodes communicate directly with the coordinator, the latter contains a TSR-bank for all the transmit nodes. For the non-invasive network all the nodes contain a TSR-bank of their neighbor nodes but majority of the communication only takes place with the coordinating node.

5.2 Related Work

With regards to MAC protocols for WBASN, various studies exist in the literature. For example various proposals from different research centers and companies were presented to *IEEE 802.15.6 TG6* (WBAN Task Group) in [75]. Similarly [120] presents an improved *IEEE. 802.15.4* MAC protocol with an adjustment of beacon traffic according to

packet arrival rate. The presented scheme performs better in terms of energy consumption than classical *IEEE. 802.15.4*. Time division multiple access (TDMA) based MAC protocols are generally considered as the most suitable for WBASN [112]. In TDMA based schemes, time synchronization of the frame and packets cost significant delay and overheads which results in extra energy waste. Each time slot is separated with *guard time* which is necessary to avoid clock drifts but this costs significant delay in real-time high data rate constraints. Moreover, the variability due to the huge traffic variations of different physiological signals being observed demands complex synchronization and extra overheads. Another TDMA-based energy efficient MAC protocol for WBASN presented in [93] is based on centrally controlled synchronized protocols. In this protocol, a master-slave based network topology is used, where master is synchronized with slave based on time slots for fixed traffic. However, in [112] after receiving a synchronization each node can transmit over multiple time slots to increase the throughput and energy efficiency, but it demands two levels of synchronization (i.e. frame and packet) which is costly in terms of delay and energy. For example, the *guard time* required for packet synchronization is 4 ms in [112], which is significant in high rate signals such as *medical images*, *EMG*, *EEG*, etc.

Very few papers in the literature explore the application dynamics in context with variation in the traffic rate for different sensor nodes connected with the body. There exist significant variations in the data rate of the same monitored parameter such as ECG, EEG, body temperature, blood pressure, etc., depending upon the health state of a patient. [9] presents a scenario-based traffic model and gives details of variation in the traffic of human body signals through ten different references which reinforce the strong need of a traffic-aware dynamic MAC protocol that can adapt a sensor node in an energy efficient manner. In this regard, [48], which is one of the proposal presented to WBAN Task Group, proposes a traffic-based secure WBAN MAC protocol. This proposal considers three traffic levels i.e. normal, on-demand and emergency traffic. Further the body network coordinator (BNC) adjusts its wake-up schedule based on wake-up patterns from low traffic to high traffic and vice versa. Moreover, it utilizes wake-up radio for on-demand and emergency traffic. This proposed technique leads to extra hardware cost by adding wake-up radio and it does not specify any mechanism for the coordinator to adapt to the variable traffic. Therefore, there exists a strong requirement for a novel energy efficient traffic-aware MAC protocol design for WBASN to deal with all types of

variable traffic that can reduce the energy consumption and improve the latency.

5.3 Applying TAD-MAC for WBASN

In the context of ultra low power and energy efficient MAC protocol, TAD-MAC is an important addition in adaptive low duty cycle MAC protocols. TAD-MAC protocol optimizes the wakeup schedule of the sensor nodes and it is designed in such a way that it enables the wake-up schedule of the sensor nodes to converge towards a coherent pattern, which is a key property of self-organization. In TAD-MAC every node adapts its wake-up interval (I_{wu}) dynamically with due account of the amount of traffic (i.e. data packets) it receives and consequently optimizes the energy consumption by waking up efficiently. The coordinator converges towards the best I_{wu} for each transmit node based on TSR. Important design parameters are the length of TSR, convergence speed and initial wake-up interval I_{iwu} value. Furthermore, the best values (in terms of convergence rate) for the length of TSR and initial wake-up interval are presented for variable traffic patterns.

Let us consider a typical body area network with five nodes connected on the body including a coordinating node. The coordinating node acts as a receive node for all the sensor nodes considered as transmitter (Tx) and connected on the body to relay the information to the local or remote monitoring station. It uses the traffic status register bank to adapt its wake up interval according to the traffic of the every transmit nodes. The status register bank includes individual status registers corresponding to each transmit node and accordingly the coordinating node evaluates different wake-up intervals for each transmit node. Moreover, the selection of a specific transmit node (for the beacon transmission) is calculated at the coordinator based on the immediate next wake-up-time of a node among all the transmit nodes.

The algorithm (presented in Sec. 4.4.3) is designed such that as it converges to a steady state value of wake-up interval the TSR contains a sequence of [10101010...] pattern as explained in previous chapter.

Fig. 5.3 shows the adaptation of the receive node wake-up interval towards a steady state value for four transmit nodes. It can be seen that there exists an oscillation pattern before reaching a steady state. This is due to the fact that during the evolution phase the receive node receives multiple data from the same transmit node (which means multiple

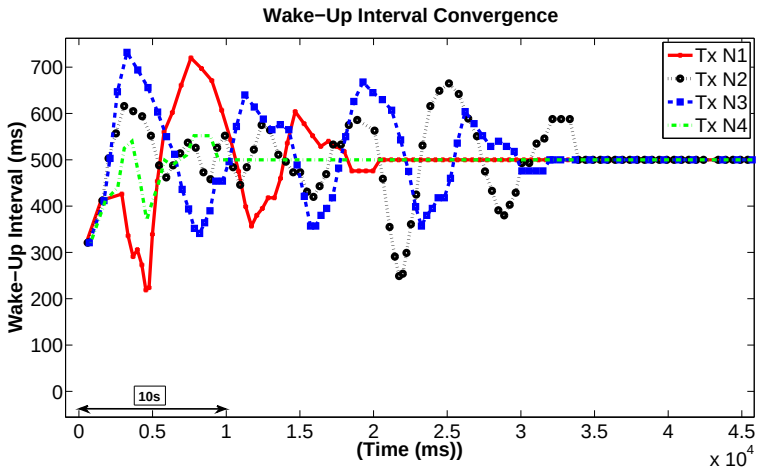


FIGURE 5.3: Receive node adaptation with respect to transmit nodes based on the wake-up interval for a fixed data rate i.e. 1 s. It is to mention here that the reference clock's increment is 1 (ms), therefore all the time values are in ms.

consecutive ones in the TSR), at that instant the wake-up interval value reduces significantly. Similarly, during the evolution phase, the receive node also receives multiple pair of zeros in the TSR, which results in an increase of the wake-up interval. After a certain number of wake-ups, the receive node converges to a steady state. Fig. 5.3 shows that the wake-up interval for all the transmit nodes converges to 500 ms within 35 s for a fixed traffic rate of 1 s.

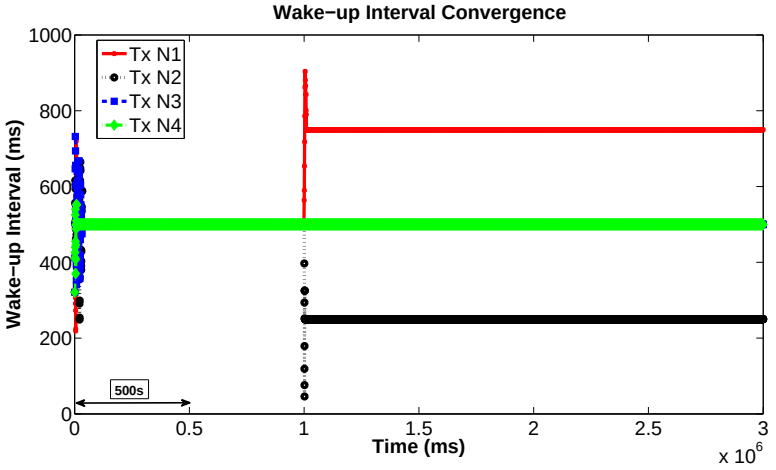


FIGURE 5.4: Convergence of wake-up interval towards a steady state value for variable sensing rates. The traffic rate for node 1 and node 2 changes from 1 s to 1.5 s and 500 ms respectively.

Fig. 5.5 which is the zoomed version of Fig. 5.4 shows the convergence of the wake-up interval for the different traffic rates (one example is highlighted in these two figures but various variations of the traffic rates are simulated which are presented later in Table 5.1). In Fig. 5.4 multiple transmit nodes first converge to $500ms$ and later two different rates

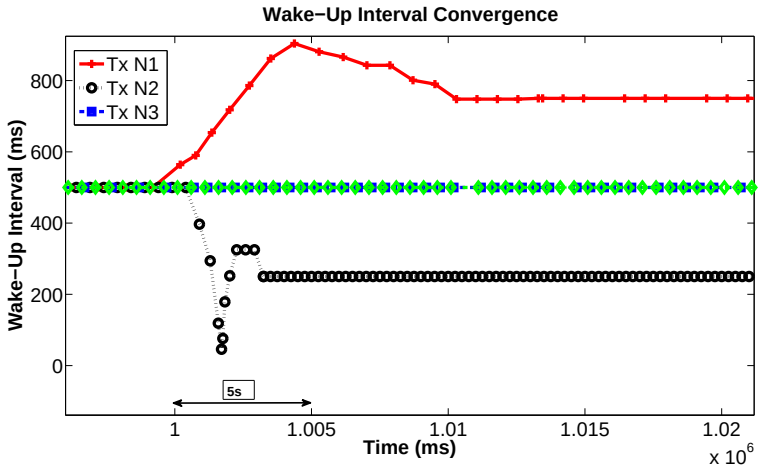


FIGURE 5.5: Zoomed version of Fig. 5.4 to show convergence after changing the traffic rate. It can be seen that the receive node converges very efficiently as soon as the traffic changes. In this particular example, the traffic rate for node 1 and node 2 changes from 1 s to 1.5 s and 500 ms respectively. Node 1 converges to a steady wake-up interval value of 750 ms, whereas node 2 converges to 250 ms for the traffic pattern of [10101010...].

occur. The traffic rate for node 1 and node 2 changes from 1 s to 1.5 s and 500 ms respectively whereas, for node 3 and node 4 sensing rate remains fixed i.e. 1 s during the entire simulation. The traffic rates expressed in Tab. 5.1 are an indication of various application scenarios that range from low to medium traffic variations. The proposed adaptive technique converges to steady state values for all the traffic loads. *TSR Length* is an optimization parameter, whereas *initial wake-up interval value* and the weighting parameter α are the two tuning parameters. After several simulations of varying traffic and by using α variations from (0.1 to 1.0), the best value of α (in terms of convergence) is found to be 0.7 (which remains constant during different traffic variations). Rest of the optimization parameters are kept fixed while evaluating the best value for α. It is important to mention that by assigning α higher than 0.5 means more weight for latest traffic in comparison with the old traffic and therefore suitable value has to be higher than 0.5. It is also found that the α values 0.5 and 0.9 does not converge to some traffic variations. Therefore, 0.7 is best value of α as it converges efficiently towards various traffic.

Table 5.1 shows the results of three different patterns of traffic variations that can be from normal to very light traffic, and vice-versa. It is found during the simulations that the initial wake-up interval value and the length of TSR are important parameters for converging fast towards a steady state. The data packet contains a field called *data-type* which gives the information about the traffic rate (whether it is burst, normal or very low)

and accordingly as the traffic switches, the receive node adjusts the length of the TSR. In this regard, a low initial wake-up interval value or a value close to half of the sensing rate seems to converge efficiently with fast convergence speed (as have been shown in Fig. 4.11 and explained in Sec. 4.5.3). The *TSR length* is another important parameter of the proposed technique. Fig. 5.6 shows the comparison between TSR length and the corresponding convergence speed under increasing and decreasing traffic variations. Generally, if the length is too short, the convergence speed is faster, but in the case of variable traffic rates, the chance of convergence reduces. On the other hand, a too long TSR takes much longer time to converge which is more energy consuming. Table 5.1 presents the best possible values in terms of fastest convergence for variable traffic rates.

TABLE 5.1: The length of the traffic status register is optimized according to the fastest convergence speed towards a steady state value of the wake-up interval for variable traffic.

Variable Traffic Packet Every(s)	Convergence Speed (No. of Wake-Ups)	TSR Length
0.5 - 1.0 - 1.5	7 - 14 - 16	4 - 6 - 8
0.5 - 1.0 - 2.0	7 - 14 - 18	4 - 6 - 8
0.5 - 1.0 - 10.0	7 - 14 - 38	4 - 6 - 12
1.5 - 1.0 - 0.5	17 - 4 - 16	6 - 6 - 4
2.0 - 1.0 - 0.5	15 - 10 - 16	6 - 6 - 4
10.0 - 1.0 - 0.5	38 - 10 - 16	10 - 6 - 4

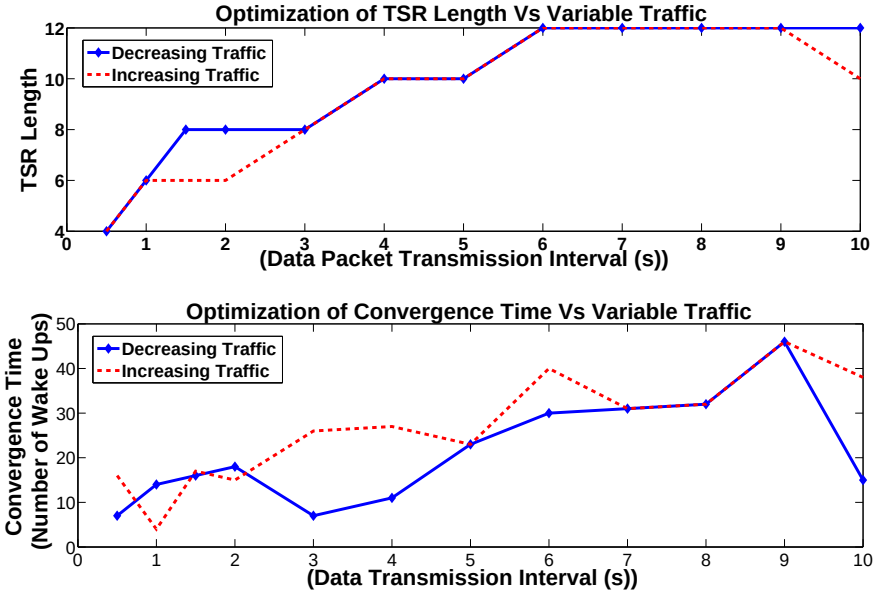


FIGURE 5.6: Optimization of the TSR length and the corresponding convergence speed for different data transmission rates, starting from two packets per second to one packet in ten seconds. It can be seen that the convergence speed increases as the rate reduces, this is due to the fact that with reduced rate TSR is filled slowly and it results in slow convergence speed.

It is worth mentioning that the above results after convergence to a steady state value consider zero energy consumption due to idle listening but for the real-time physical implementation there will be very little energy waste due to clock drift and hardware latencies. Also, the energy consumed by the wake-up beacon transmission is reduced significantly as number of unnecessary wake-up beacons are avoided through the proposed technique. The latter can be applied to both transmitter initiated and receiver initiated MAC protocols of preamble sampling category for various applications.

5.4 Performance Evaluation

In this section the energy estimation of different energy efficient MAC protocols is presented. The results are split into fixed and variable traffic rates, further the energy consumption of three different widely used radio chips are explained. Based on the energy consumption of different MAC protocols and radio chips the lifetime of the sensor nodes is presented.

Table 5.2 illustrates the important parameters of three different widely used radio chips of WSN/WBASN, these being *cc*2420 [55], *amis*52100 [17] and *cc*1000 [54]. With regards to the energy consumption, clearly *amis*52100 and *cc*1000 are designed specifically for protocols that spend more time in receive states (such as receiving the data or waiting to receive the preambles, waiting for the beacon or acknowledgment, etc.). Whereas, *cc*2420 consumes significantly more power in the receive state but the transmit power is much less than the other two chips. Another important contrasting design feature is the operating frequency band. *cc*2420 chip is designed for higher spectrum ISM band i.e. 2.4 GHz and it also has much higher data rate capabilities in comparison with the other chips. Whereas, *cc*1000 and *amis*52100 operate at lower frequency bands such as 433 MHz ISM band, MICS (medical implant communications service) and WMTS (wireless medical telemetry service) bands.

5.4.1 Energy Consumption Evaluation for Fixed Traffic

In order to evaluate the energy consumed of various radio chips, it is necessary to analyze various MAC protocols that actually control the functionality of the radio chip. In this regard, for a static network and fixed traffic, low power and energy efficient preamble

TABLE 5.2: Parameter specifications of three widely used radio chips for WSN/W-BASN.

Parameters	**cc2420**	**amis52100**	**cc1000**
Modulation Schemes	OQPSK	ASK/OOK	FSK/OOK
Sensitivity (dBm)	−95	−117	−109
Size(mm)	7.0×7.0	7.5×7.8	9.7×6.4
Data Rate (Kbps)	250	19	76.8
Clock Drift (ppm)	40	80	50
Maximum Transmit Current (mA)	17.0	25	26.7
Maximum Receive Current (mA)	19.6	7.5	7.4

sampling MAC protocols (presented in Sec. 2.2) are selected for comparison with the presented protocol. These protocols are used in majority of the WSN motes for applications ranging from health care, environment monitoring to surveillance and many more.

The network simulator *WSNet* [27] [28] is used to evaluate the time spent by different MAC protocols in various states. The simulation setup consists of 10 nodes including a coordinating node based on a star topology. During the simulation, it is assumed that the network traffic follows Poisson distribution and uses a rayleigh fading channel model. Based on the application model, each node senses the channel with a fixed rate of 100 ms and the data rate is considered as 1 packet/s. The simulation runs for 50 minutes. For the selected protocols, i.e. BMAC, XMAC, RICER, WiseMAC and TAD-MAC, the average time spent by transmit and receive nodes are evaluated as shown in Table 5.3. Furthermore the energy consumption of these different MAC protocols for three different radio chips is also provided.

In BMAC protocol (a detailed description of the protocol can be found in [95]), intended transmit node appends a long preamble ahead of the data such that the destination node must receive the preamble as it wakes up. There is a significant energy waste due to long preamble in the form of idle transmission (at Tx Node) and overhearing (at Rx Nodes). The time spent by the transmit node in *Tx* state is 0.7 ms and 110 ms in *Idle* state, whereas the receive node spends 11 ms in *Rx* state. It can be observed that actual transmission and reception times are very less in comparison to the *Idle* state. Total energy consumptions by various radio chips for BMAC protocols, shown in Table 5.3,

are evaluated by taking the maximum current levels of the chips (provided in Table 5.2) multiplied with the time evaluated for different states and a constant 3 V battery.

TABLE 5.3: Energy consumption (for one complete communication) evaluated for different radio chips and for different MAC protocols. The time spent by protocols in *Tx*, *Rx* and *Idle* states are evaluated through *WSNet* network simulator. Further by using the characteristics of different radio chips (shown in Table 5.2) energy consumption is estimated.

	Radio Chips	States	BMAC [95]	XMAC [24]	RICER [82]	WiseMAC [60]	TAD-MAC
Time (ms)		Tx	0.70	0.0128	0.73	0.71	0.73
		Rx	11.0	0.0128	0.73	0.04	0.70
		Idle	**110.0**	**100**	**51.3**	**50.21**	**2.56**
		Total	121.7	100.07	52.7	50.9	3.99
Energy (mJ)	cc2420	Tx	0.035	0.000652	0.0376	0.038	0.037
		Rx	0.646	0.000752	0.0433	0.045	0.043
		Idle	**5.610**	**5.102**	**3.01**	**0.56**	**0.15**
		Total	6.291	5.104	3.096	0.648	0.230
Energy (mJ)	cc1000	Tx	0.056	0.00102	0.059	0.059	0.059
		Rx	0.244	0.000284	0.016	0.016	0.016
		Idle	**8.811**	**8.014**	**1.138**	**0.57**	**0.056**
		Total	9.111	8.015	1.214	0.66	0.131
Energy (mJ)	amis52100	Tx	0.052	0.000960	0.055	0.055	0.055
		Rx	0.247	0.000288	0.016	0.016	0.016
		Idle	**8.250**	**7.503**	**1.154**	**0.57**	**0.057**
		Total	8.549	7.505	1.225	0.653	0.128

XMAC [24] improves the energy waste by replacing the long preamble with multiple short preambles transmitted at short intervals and as soon as the transmit node receives the acknowledgment from the destination, it transmits the data. With these optimizations, XMAC reduces the idle transmission cost as well as the overhearing cost. The transmission and reception time values are 0.0120 ms and 0.0128 ms respectively, whereas the idle transmission time is 100.052 ms.

RICER [82] is a receiver initiated protocol in which the transmit node waits for the wake-up beacon from the destination before it transmits the data. The transmit node spends maximum energy in idle listening (for waiting the wake-up beacon). It is important to note that the radio chips *amis*52100 and *cc*1000 consume less energy in comparison with *cc*2420, this is due to the fact that the current level of *amis*52100 and *cc*1000 is much lower than *cc*2420.

WiseMAC is another energy efficient MAC protocol in which transmit node adapts itself according to the wake-up schedule of the receive node. At the very first communication

the receive node informs the transmit node about the next wake-up time in the *acknowledgment*. WiseMAC contains a very short preamble which is calculated based on the formula $4\theta T_L$, where θ is the clock drift and T_L is the time elapsed since the last communication. Further, it also includes *medium reservation preamble* to avoid collision by introducing a short random offset (the detailed description of the protocol can be found in [60]). It is important to note that since the transmit node waits until the receive node wakes up (as can be seen from Fig. 2.5), it sleeps the radio component until the time close to the receive node wakes up, which eventually avoids the unnecessary idle energy waste. On the other hand, the microprocessor continues to remain in the active state (to make sure that the radio is turned on at the right moment), though it consumes much less energy in comparison with the radio being in receive state (idle listening). It is to mention here that in all the other protocols (which means in different states of the protocol) the energy consumption of MCU is approximately the same [14].

TAD-MAC is a dynamic version of a receiver initiated protocol which is adaptively adjustable to various traffic variations. For fixed traffic, TAD-MAC also improves the energy waste due to idle listening and, as the receive node converges towards the steady state according to the transmit node, the idle listening is present only to mitigate the effect of clock drift and hardware latency. The energy consumption is less than 1 mJ for *cc*2420 and is less than 0.5 mJ for *amis*52100 and *cc*1000. The time spent by the transmit node in idle listening is only 0.7258 ms which helps to reduce the important energy consumption due to idle listening.

Lifetime estimation of the battery and its improvement is the ultimate design objective in WBASN. Fig. 5.7 shows the lifetime estimation for the different MAC protocols. The total available energy is estimated from the capacity of two AA alkaline batteries provided by energizer company with 1.5V nominal voltage. [70] shows the current and voltage discharge profiles for various operations and we have considered the battery capacity equals 3124.2 mAh. The Lifetime is defined as the time when first node reaches out of energy. The bar graph presented in Fig. 5.7, clearly shows that the TAD-MAC with *amis*52100 or *cc*1000 radio chips has huge difference in comparison with other MAC protocols and *cc*2420. The Lifetime using TAD-MAC protocols is estimated to be equal to *2100* days, *3400* days and *3500* days for *cc*2420, *cc*1000 and *amis*52100 chips respectively. TAD-MAC has 3 to 18 times more battery lifetime in comparison with

other protocols. It is noteworthy to mention here that for low traffic rate *amis*52100 and *cc*1000 are preferable but for higher traffic rate *cc*2420 is more suitable.

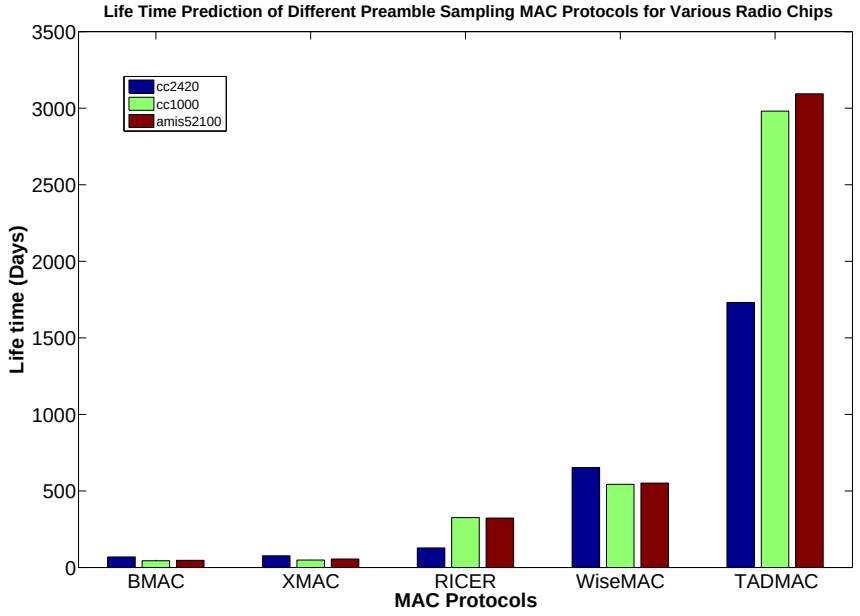


FIGURE 5.7: Lifetime prediction of AA alkaline battery is evaluated for various MAC protocols and radio chips. TAD-MAC along with *amis*52100 or *cc*1000 radio chips have major difference in comparison with other MAC protocols and *cc*2420. It is noteworthy to mention here that for low traffic rate *amis*52100 and *cc*1000 are preferable but for higher traffic rate *cc*2420 is more suitable.

5.4.2 Energy Consumption Evaluation for Variable Traffic

In this context of WBASN variable traffic, it is very important to consider a realistic comparison for different MAC protocols. The protocols that are used for the fixed traffic can not be used for comparison with our TAD-MAC protocol for the case of variable traffic because all the protocols are designed with fixed wake-up interval. BMAC, XMAC and RICER suffers from packet losses, latency and huge energy consumption, whereas, WiseMAC, which can be considered as a traffic-aware protocol, yet needs significant modifications to be able to be applied in WBASN.

There are two important issues related with WiseMAC with regards to variable traffic of WBASN. If the wake-up interval is kept fixed but its value is selected based on the fastest rate of physiological data, in this case, the nodes which are used to send body temperature and heart rate will wake-up unnecessarily tens and hundreds of times more than they need which will result in energy waste. On the other hand, if each node has a

different wake-up interval (which is known among the neighbor nodes), according to the amount of packets a node transmits and receives, as soon as there will be variations within one physiological data, the transmission rate demands the change in wake-up interval which will not be possible as the wake-up intervals are different but fixed. Lastly, if each node allows to change its wake-up interval at run time, WiseMAC can update the new wake-up interval of the receive node but only a specific transmit node will be able to have that update, whereas the other nodes will still have the old wake-up interval value. [62] presents an extended WiseMAC to improve the throughput but it does not address the above issues.

TAD-MAC results for variable traffic are presented in Table 5.5. Four different physiological data (as shown in Table 5.4) are transmitted to the coordinator node, these include *ECG-compressed* (10 packets/s), *oxygen saturation* (1 packet/s), *Temperature* (1 packet/s) and *Heart Rate* (2 packets/s). The transmission and reception time values are 0.73 ms and 0.71 ms respectively, whereas the idle transmission time is 3.2 ms. These time values are the average values evaluated based on the simulation run of 50 minutes in *WSNet*.

TABLE 5.4: Physiological data being monitored and their respective data rates.

Monitored parameters	**Transmit Packets/sec**
ECG.	10
Oxygen saturation	1
Temperature	1
Heart rate	2

TABLE 5.5: Energy consumption of the TAD-MAC protocol under variable traffic of WBASN for three different radio chips. The energy consumption for one complete communication is less than 0.3 *mJ* for *cc*2420 and is less than 0.2 *mJ* for *amis*52100 and *cc*1000. The time spent by the transmit node in idle listening is only 3.2 *ms*.

States	**cc2420 (Joules)**	**amis52100 (Joules)**	**cc1000 (Joules)**
Tx	3.76×10^{-5}	5.52×10^{-5}	5.90×10^{-5}
Rx	4.33×10^{-5}	1.65×10^{-5}	1.63×10^{-5}
Idle	0.18×10^{-3}	0.07×10^{-3}	0.07×10^{-3}
Total	0.26×10^{-3}	0.14×10^{-3}	0.14×10^{-3}

By using the results presented in Table 5.5, the lifetime using the TAD-MAC protocol is estimated to be equal to *1765* days, *3100* days and *2980* days for *cc*2420, *cc*1000 and *amis*52100 chips respectively.

Finally, a small comparison with energy efficient low duty cycle TDMA-based protocol [112] is presented. It considered a TDMA frame of 1 s, packet size of 1250 bits with 80 bits of overhead sampling rate of 125 samples/s. It achieves 4.51 % duty cycle but each node requires 48.3 ms to completely transmit its data packet. This total time includes; *packet transmit*, *waiting time*, *ACK. received* and *guard time*, which is much more in comparison with our TAD-MAC protocol under *ECG* transmission at 1.2 kbps.

5.5 Conclusion

TAD-MAC protocol is an energy-latency optimized MAC protocol which relies on real-time traffic estimation based on a traffic status register used to compute an adaptive algorithm. In this chapter TAD-MAC is applied in the context of WBASN. A body area network with hybrid topology is considered, which is suitable for both invasive and non-invasive BAN. Firstly, the simulation results of fixed and variable traffic are presented, it is shown that the wake-up interval converges efficiently to a steady state value for different traffic rates. The *TSR* length is adjusted as an optimization parameter to obtain fast convergence and its values for various variations of traffic rates are presented. Secondly, performance evaluation of different energy efficient preamble sampling MAC protocols for fixed and variable traffic rates is presented.

Our proposed dynamic MAC protocol outperforms all the other protocols in fixed and variable traffic. Moreover, the energy consumption of three radio chips using TAD-MAC have clear difference in comparison with other protocols implementation. For the fixed traffic, the closest comparison is with WiseMAC, for *cc*2420 radio chip TAD-MAC has 3 times more lifetime than WiseMAC. Whereas, TAD-MAC has 6 time improvements by using *cc*1000 and *amis*52100. For variable traffic, there is a significant degradation in performance of non-dynamic protocols with regards to energy consumption, quality of service and latency. To be able to show a fair comparison there is a need of modifications among other protocols. Thus, a discussion of several modifications that are necessary before we can perform a fair comparison is presented.

Chapter 6

Adaptive Transmit Power Optimizations for WSN

Optimizing the energy consumption due to idle listening, overhearing, collisions and overheads at the MAC layer helps significantly to improve the lifetime of the network. Moreover, the transmission of control packets(i.e. beacon) is also minimized. A dynamic adaptation of the MAC layer results in a reduction of both transmit and receive nodes energy consumption through above mentioned optimizations. In this chapter we will focus on radio output transmit power which is the second important factor that contributes to the transmitter energy consumption at the physical layer and we will introduce a dynamic adaptation of the transmit power under varying channel environment to further reduce the energy consumption of WSN.

6.1 Introduction

In the context of radio transmit power optimization, connectivity between the nodes is a major concern. Transmit power can be reduced if the signal strength is better than environment noise. To ensure reliable and effective communications, low transmit power can be adopted under good channel conditions, while a higher power is necessary for bad channel conditions. In these regards, if the transmit power is too low then, there are multiple disconnected clusters of nodes instead of a single well connected network. Whereas, transmitting at excessively high power would lead to higher interference and

a shorter battery life [94]. Therefore, to obtain a good connectivity, increased lifetime and reduced interference, the transmit power must be (just) high enough for a node to communicate with any other node in the desired area.

On the one hand, it is preferable to assure the adaptation for the minimum transmission power in order to extend WSN lifetime as much as possible. On the other hand, it is crucial to guarantee that the transmitted data is correctly received by the other nodes. For example, if the transmit power is reduced too low, it will result in more energy consumption because of too many retransmissions due to errors in the data and therefore the overall energy efficiency will be reduced. Thus, trading-off between near-to-optimal power transmission and reliability of successful transmission is one of the most important concerns addressed in this chapter.

One of the application of the hybrid energy model mentioned in Sec. 3.6.2 was the static transmit power optimization, where the three-dimensional curve shown in Fig. 3.16 represents the variation of energy per successfully transmitted bit at different distances and different power levels. With regards to various low-power radio transceivers used in WSN, often there are multiple transmit power levels available that can be tuned to reduce power consumption [55], [105]. The static power optimization can perform well if all the nodes are considered stationary and environment noise is assumed to be constant. The second assumption is not practical because even for static networks (for both indoor and outdoor deployments) there are some channel variations which can significantly degrade the performance of the communication system if the transmit power is not selected to its maximum level. Therefore, the point at different distances shown in Fig. 3.16 can only be effective if the transmit power level is adapted according to the channel variations.

In this context, an adaptive transmission power control technique is presented in this chapter that is able to guarantee the successful transmission of data as well as to maintain near-to-optimal transmit power level for energy efficiency. The main idea behind the proposed approach is to adapt the transmit power level according to signal-to-noise (SNR) variations. After receiving a data packet the receive node calculates the energy per-bit (E_b) through bit-error-rate (BER) (from Eq. 3.26). The energy per-bit is calculated against number of different transmit power values. The best transmit power is obtained from minimum energy per successful transmitted bit. The energy consumption

is relatively higher for high power levels in comparison with the low levels and the optimal points are identified from a gradient of the curve. Different channel profiles are used to evaluate the performance of the adaptive technique and further the results of energy consumption are computed for single packet transmission under low, medium and high SNR values and the improvements are shown through a comparison of static transmit power with a dynamic power optimization. Finally several link quality estimation techniques are presented for implementation considerations.

The rest of this chapter is organized as follows. In the next section, related works on link quality estimation is presented followed by Sec. 6.3 the transmit power optimization techniques for WSN which includes first the performance of static transmit power. Then, various channel profiles are then presented and an adaptive transmit power control technique is proposed, which is evaluated by the simulation results. Performance is analyzed by comparing the results of adaptive transmit power technique with the fixed transmit power. Further, a discussion for possible implementation approach is presented in Sec. 6.6, followed by a conclusion..

6.2 Related Work

There are number of distributed power control algorithms reported in the literature, some are designed with network connectivity and network longevity as their objectives such as proposed in [101], [103] and [125]. An algorithm proposed in [20] attempted to balance the power and delay trade-off by using local information such as last transmitted power level, last interference level, and further, at the transmitter to determine the appropriate power to transmit in the next time slot. [42] proposed a power management scheme for improving the end-to-end network throughput.

Some of the other existing solutions for transmission power control assume that every node chooses a single configurable transmission power for all the neighboring nodes [101] [72]. Ideally, the transmit power of a node should be adjusted on a link-by-link basis to achieve the maximum possible power savings [8], [101]. [94] has reported that, due to the absence of a central controller in a pure ad hoc network (such as WSN) with flat architecture, performing power control on a link-by-link basis is a complicated and cumbersome task. A simpler solution, is to have all the nodes using a common (fixed)

transmit power. However, this solution is not energy efficient because transmit power can be optimized depending upon the quality of the link.

Most of the above mentioned power control algorithms were designed with the objective of ensuring network connectivity without considering any appropriate link quality estimation (LQE) metric. Below a state-of-the-art on link quality estimation techniques is presented which will guide us for the selection of LQE metric for adaptive transmit power optimization.

6.2.1 Link Quality Estimators

Radio link quality estimation (LQE) is a fundamental building block for wireless sensor networks, namely for a reliable deployment, resource management and routing [51]. Adapting the transmit power under varying channel conditions requires an accurate link quality estimation. Generally link quality estimations can be divided into hardware and software estimators. Hardware-based estimators utilize built-in parameters of the radios such as received signal strength indicator (RSSI), link quality indicator (LQI), signal-to-noise ratio (SNR). These estimators do not need any extra cost as the parameters are part of receiver hardware. Whereas, the software-based estimators either count or approximate the packet reception ratio or the average number of packet transmission or re-transmission. These estimators are packet reception ratio (PRR), required number of packets re-transmissions (RNP), expected transmission counts (ETX) and window mean exponential weighted moving average (WMEWMA). Below, different metrics of link quality hardware based estimation methods that are relevant to our proposed technique are presented.

6.2.1.1 SNR-Based Link Quality Estimation

Signal-to-noise ratio is an important metric to characterize and measure a link quality. [85] presents an interesting insight of link quality estimation for a static network by using this SNR metric. The computation of SNR is achieved by sampling the received signal strength indicator (RSSI) for every byte of data a node receives. Then, it averages these values over an entire packet length, it calculates an estimate of the average received signal power for a packet. By further sampling after the packet reception was complete, the

receiver also obtained a value for the average noise power immediately after the packet. The receiver had a priori knowledge of the packet format and thus was able to compute the number of byte errors as well as single bit errors in the received packets.

6.2.1.2 Multiple Metric-Based Link Quality Estimation

On one hand, RSSI is not considered as a reliable measure for the link quality estimation because of the asymmetry links of up-link and down-link [49]. Therefore, to have accurate link estimation [49], proposed to utilize SNR in combination with packet delivery ratio, stability factor and asymmetry. The stability is defined as the coefficient-of-variation of the 30 most recent packet reception ratio (PRR) values of a link. This method of using more than one parameter to estimate link quality has been more effective than using a single metric because a single link metric accesses a particular link property and thus provides a partial characterization of the link. On the other hand, [113] reports that RSSI is under appreciated mainly due to not proper calibration of the older radios such as *cc-1000* and *TR-1000*. In more latest radios such as *cc-2420*, *cc-2500* and others, there is no problem of calibration. However, it has been shown that RSSI works well above the sensitivity threshold of the radio, whereas, averaged LQI has a better performance than RSSI under low SNR as it has a better correlation with PRR. [45] proposed a link quality estimator that combines PRR and normalized RSSI. Generally, it has been found that by normalizing or averaging RSSI over a certain length helps to achieve better estimation instead of using an absolute RSSI value. The authors in [44] also have used multiple metrics (i.e., SNR, PRR and LQI) for fast and accurate link quality estimation. [50] proposed a similar technique and shows better improvements under variable link conditions.

6.2.1.3 Chip Error-Based Link Quality Estimation

[58] presents an alternate approach for fast and accurate packet delivery estimation. It uses *chip errors* in symbols for direct sequence spread spectrum transceivers. The link quality estimator is evaluated experimentally on IEEE 802.15.4 for different link conditions, including multi-path and mobile scenarios. It has been reported that the *chip errors* based link quality estimator performs more accurately than received signal

strength based estimators and much faster than the packet statistics based estimators (i.e. PRR) with comparable accuracy.

To conclude on the various link quality estimation methods discussed above, it is observed that in case of good links with a stable PRR above 95% and with SNR well above 5 dB, all estimators achieve good performance. Whereas, in the case of bad links the performance of almost all the estimators degrades.

6.2.2 Link Quality Estimation for Adaptive Transmit Power

In the context of adaptive transmit power, it is important to have an accurate measure of link quality such that maximum energy efficiency can be achieved by tuning appropriate power level of radio transceivers and also for appropriate routing and other necessary optimizations for WSN. In this chapter we have used SNR as LQE metric which is implicitly dependent upon RSSI as the received and noise power (in practice) can be estimated from RSSI (as explained in Sec. 6.6).

6.3 Transmit Power Optimization for WSN

Transmission power control is a mechanism that aims to reduce the power consumption and increase the battery lifetime, especially in some applications where the nodes transmit too often. In many wireless sensor networks applications, fixed transmission power is used for all the nodes without taking into considerations the variable distance between the nodes in the network and the channel variations. An adaptive transmission power can be employed by the transmitter to reduce the unnecessary energy consumption. Transmit power can be reduced if the signal strength is better than environment noise. To ensure reliable and effective communication, low power is adopted under good channel conditions, while high power for bad channel conditions. This is the principle of transmission power control.

6.3.1 Power Profiles of Radio Transceivers

The power profile characteristics of different radio transceivers operating at 2.4GHz and 900MHz of spectrum (which are ISM bands widely used for WSN) are shown in Fig. 6.1.

It shows the relation between the transmit power and corresponding current consumption. There are two important points related with the radio transceivers power profile. First, all the radio transceivers shown in Fig. 6.1 have only few possible transmit power levels which means that the adaptive transmit power optimization has to make a compromise by selecting one of these power levels and not the optimal power level as for example shown in Fig. 6.3. So, in order to be always effective (i.e. energy efficient), the selected power level should be higher than that of best power level. Second, most of the radio transceivers have very flat power profile especially for lower power levels which results in a low gain in terms of adaptive transmit power optimization. It may be better for such transceivers to not operate at very low power level.

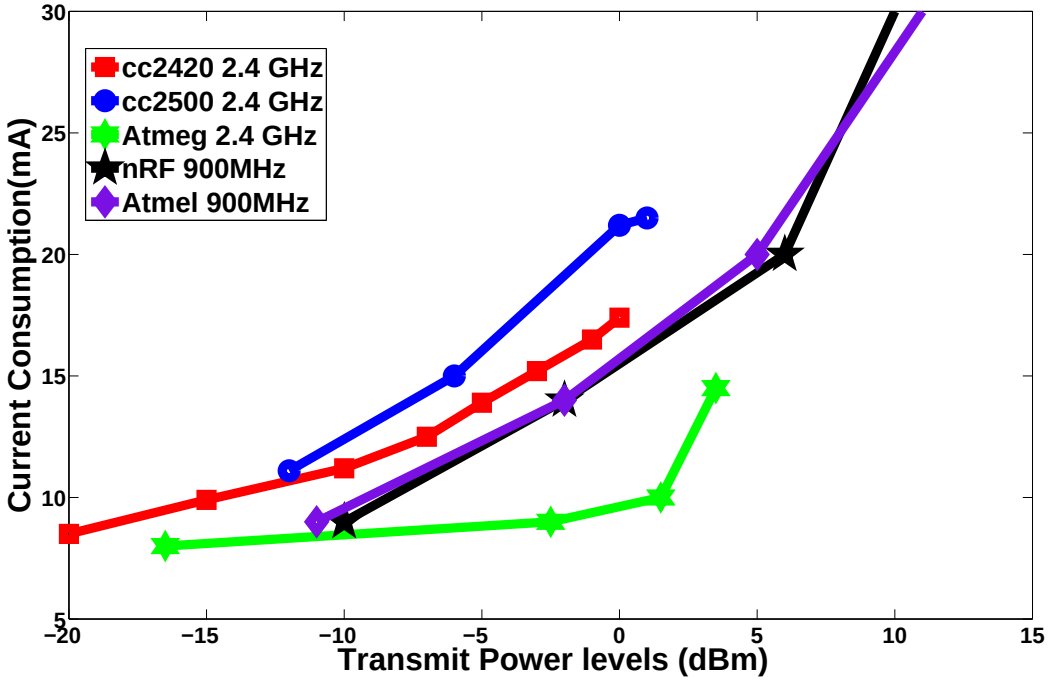


FIGURE 6.1: Power profile characteristics of different radio transceivers operating at 2.4GHz and 900MHz. The curves show the current consumption against various power levels of the radio.

It is important for adaptive transmit power optimization that there a significant/meaningful difference between each power level of the radio chip. In these regards, we have considered two radio transceivers for further analysis one from each spectrum i.e. *cc-2420* for 2.4 GHz ISM band and *nRF-900* for sub-giga spectrum. It can be seen that the radios operating at 900MHz have higher current consumption but they have longer distance range. On the other hand, the radios operating at 2.4GHz have relatively lower current consumption but they also have shorter distance range.

6.3.2 Performance of Static Transmit Power

In this section performance of transmit power optimization under static conditions such as fixed distance and fixed noise is presented. In many wireless sensor networks applications, fixed transmission power is used for all the nodes without taking into considerations the variable distance between the nodes in the network and the channel variations. On the one hand, keeping the transmit power fixed to its maximum value is useful under strong interference and at higher environment noise. On the other hand, under favorable conditions transmit power can be reduced to improve energy efficiency of WSN. In this context first, we present the relation between the energy per bit versus transmit power under static conditions.

Fig. 6.3 shows the energy consumption per successful transmitted bit for fixed noise power (i.e. -90dBm, which is considered as very low or favorable environment condition) with respect to different transmit power values. This curve is obtained through Matlab simulation. It uses an additive white Gaussian noise channel model as described in Sec. 3.2.4, with a channel attenuation coefficient γ equal to two, distance is considered fixed as ten meters, acceptable BER (bit-error-rate) is considered as 10^{-3} for a modulation scheme of BPSK and the energy per bit is estimated from

$$E_b = \frac{P_{tot} T_{tot}}{TX_{packets} pl} \tag{6.1}$$

where P_{tot} is the total power being consumed (including amplify power, digital and analog system power), and T_{tot} is the total time to transmit successfully all the packets. $TX_{packets}$ and pl are the total number of packets to be transmitted and the packet length respectively. By decomposing the expression of T_{tot} based on number of automatic repeat request (ARQ) re-transmissions and using Eq. 3.1, the final expression for E_b becomes

$$E_b = \frac{P_{tot}}{R(1 - BER)^{pl}} \tag{6.2}$$

where R is transmission bit rate. During all simulations $TX_{packets}$ and pl are considered as 1000 packets and 16 bytes respectively as an example.

Fig. 6.2 shows the energy consumption of successful transmitted bit and corresponding BER for number of different transmit power levels under fixed noise. It can be observed

that, as the transmit power reduces the BER starts reducing, and the energy consumption starts increasing. In Fig. 6.2, the transmit power level around −17dBm provides the minimum energy per bit and at the same time satisfies the minimum BER (i.e. 10^{-3}), and therefore can be considered as the best power level for the given noise level.

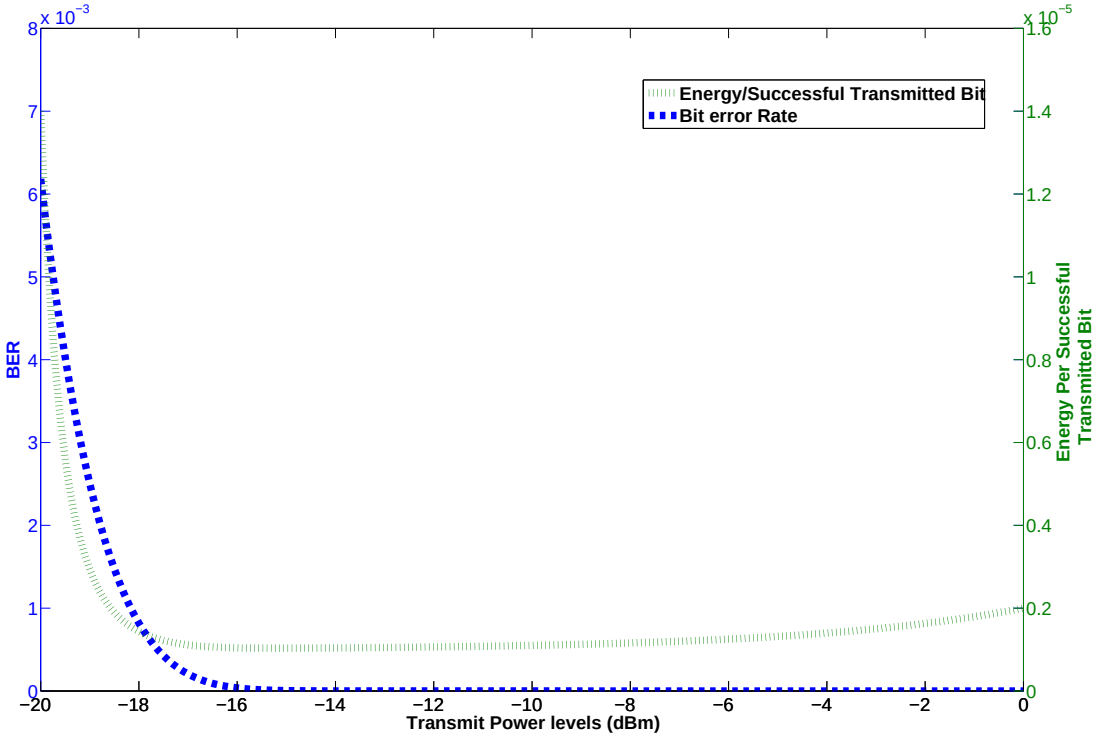


FIGURE 6.2: Energy per successful transmitted bit and corresponding bit error rate versus different transmit power levels under fixed noise (-90dBm).

The curve (Fig. 6.3), shows three different regions, first as the power level starts reducing (from the maximum power level) the energy consumption reduces exponentially. In the second (plateau) region it can be seen that reducing the transmit power only has very minor reduction in energy consumption. Whereas, in the third region it can be seen that further reducing the transmit power results in an exponential increase of energy consumption. This is due to the fact of retransmission and error detection because of very low power level.

It is clear from Fig. 6.3, that the transmit power can be reduced instead of keeping it fixed at maximum power level. However, it is important to note that reducing the transmit power to a very low level can result in degrading the energy efficiency due to re-transmission as most of the packets are lost or received with errors. Therefore, the best transmit power has to be carefully selected such that the energy consumption remains minimum even with slow channel variations. It is important for an adaptive transmit

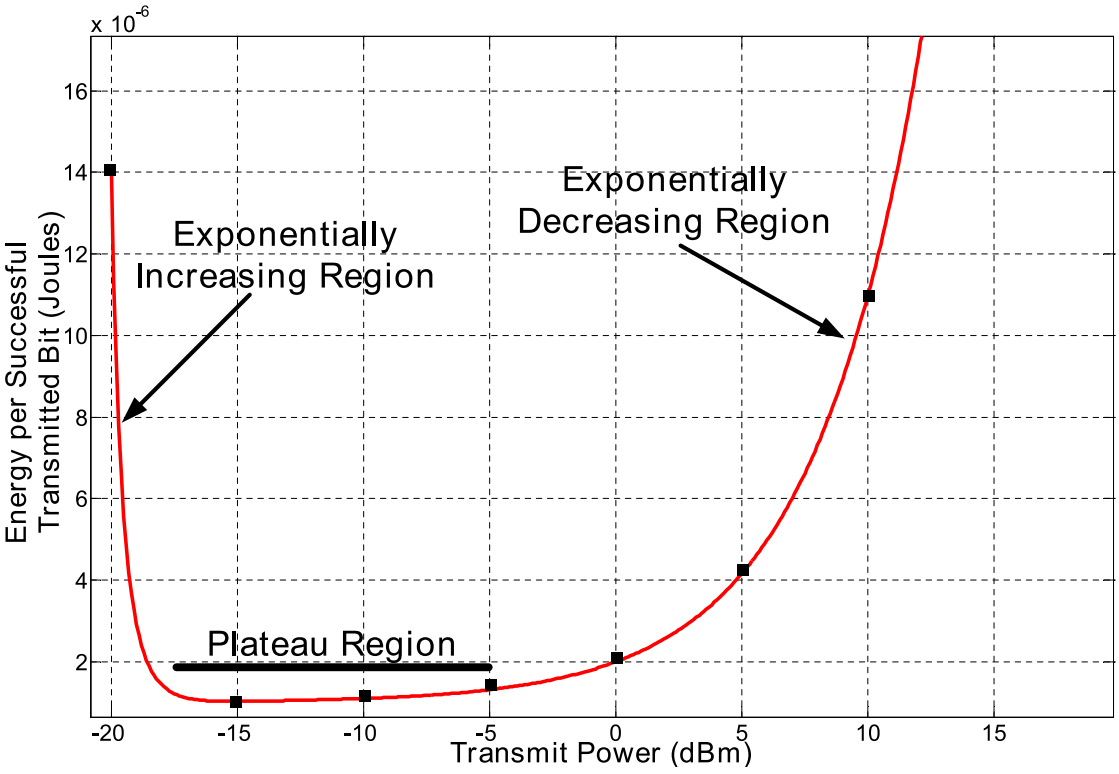


FIGURE 6.3: Energy consumption under fixed noise power (–90dBm) against different transmit powers.

power optimization that there is a significant/meaningful difference between each power level of the radio chip. In these regards, we have considered two radio transceivers for further analysis one from each spectrum i.e. *cc-2420* for 2.4 GHz ISM band and *nRF-900* for sub-giga spectrum.

In the next section a channel profile is presented which will be later used in the adaptation of the transmit power.

6.3.3 Channel Profiles

Channel model is one of the important factor to evaluate the theoretical performance of adaptive transmit power control technique (which is presented in Sec. 6.4). We have considered an additive random noise channel profile to verify the performance of the adaptive algorithm. First, the randomness is selected according to different scenarios of obstacles such as slow and fast moving objects in the propagation path. Second, the variations are selected by knowing the fact that radio transmitter has a limit to maximum transmit power and if the noise power exceeds a certain limit then the energy

consumption will increase alarmingly. Therefore, we have considered different flat-fading slow-varying channel profiles which are shown in Fig. 6.4.

First of all different path losses (as explained in Sec. 3.2.4) are calculated at the receiver and then signal-to-noise ratio (SNR) is computed at the receiver over fixed distance. It is important to mention here that we can also consider variable distances and adapt the transmit power based on the distance matrix (as for example presented in [129]), but as the computation of SNR is also based on the distances, therefore, our analysis implicitly includes distance as well but not as a separate parameter. Moreover, relying the computation on the SNR rather than on the distance is a smart approach since depending on the environment there could be huge variations in the link quality for different nodes at the same distance.

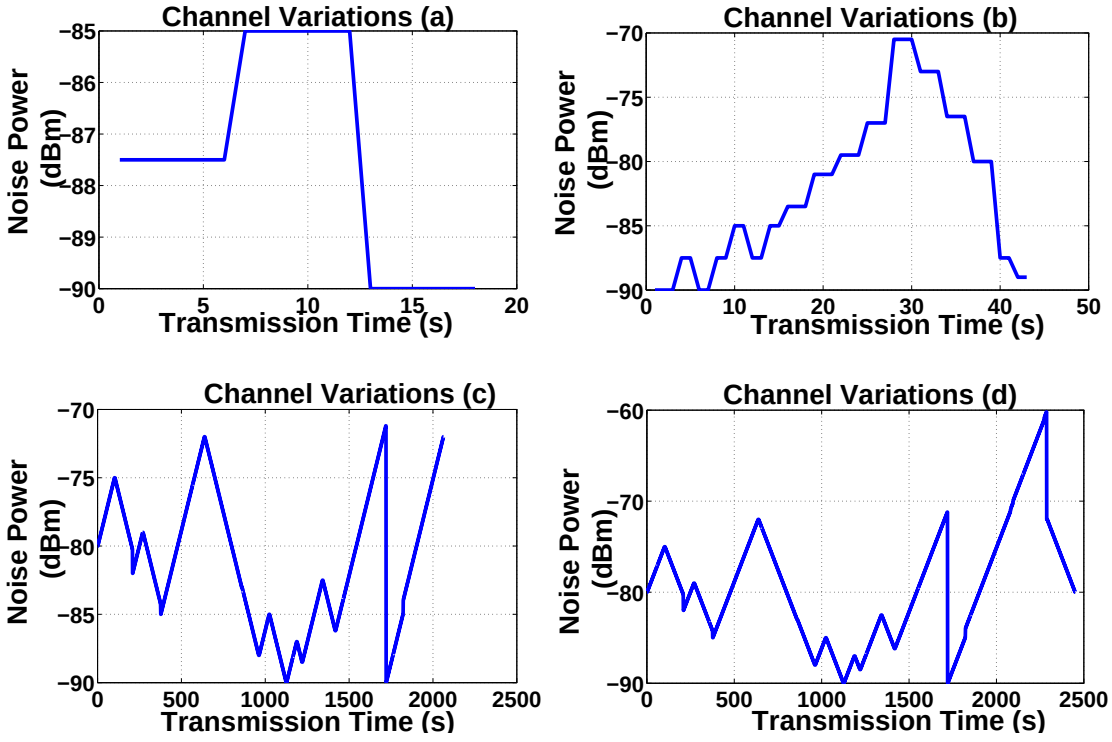


FIGURE 6.4: Different variants of varying noise power based Channel profiles. The limits of variations of the noise power are according to the maximum power that is available in most widely used radio transceivers for WSN.

Four noise power profiles are considered in our experiments as shown Fig. 6.4. First, a step-based additive noise (Fig. 6.4-a) is used which contains fixed noise power for number of data packets and allows the adaptive algorithm to obtain the best transmit power level and then the noise power is changed to another fixed value. This is the first step to see if the algorithm adapts to the optimal transmit power effectively or not.

Second, number of different steps are considered with more variations in comparison to the first model as shown in Fig. 6.4-b (it is also used in Fig. 6.5), though it is also mainly either increasing the noise power or decreasing it with a unique flow. Third, more random noise power variations are considered to guarantee the performance of the adaptive algorithm. Therefore, Fig. 6.4-c shows extreme noise variations by precisely keeping in mind the maximum power available in radio transceiver *cc-2420*. This radio transceiver has multiple transmit power levels steps from very low (i.e. −25dBm) until 0dBm. In this model only two packets are assumed to have same channel conditions which means the channel is varying faster than for *model-a* and *model-b*. Finally, *model-d* is considered for *sub-giga* transceivers which normally operate at 900 Mhz or even lower frequencies. Normally the radio transceivers which operate at lower frequencies have higher transmission range or in other words higher transmit power levels and more distance range. So, *model-d* is targeted for *sub-giga* radios and it has higher noise power in comparison to *model-c*.

6.4 Adaptive Transmit Power Technique

When looking at the static conditions which means fixed noise and distances between the nodes, it is evident that there exists a point on the curve which provides the minimum energy consumption against the best power level. In practice, it is not possible to have a fixed noise power and constant SNR therefore, often the transmit power is kept fixed to its maximum value which is not an optimized solution especially for limited battery operating devices.

Fig. 6.5 shows the energy per successful transmitted bit versus different transmit powers under varying noise power level of *model-b*. Each curve represents different but fixed noise power on different curves are obtained at the optimal transmit power levels (in terms of energy) on each curve. It can be seen that for low noise power the transmit power can be reduced in order to optimize the energy consumption as the best point on the curve is located at the lower power levels. Whereas, for high noise only higher values of the transmit power can be selected in order to be energy efficient (which means to avoid the sharp increase in the energy consumption as can be seen in all the curves).

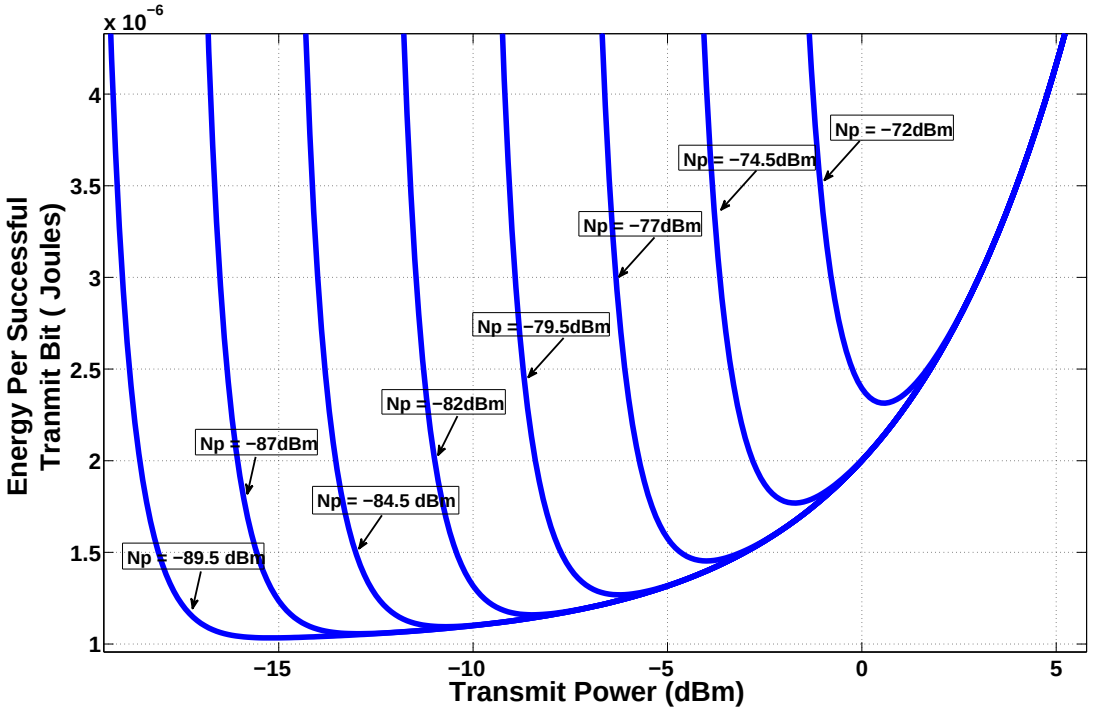


FIGURE 6.5: Energy consumption versus transmit power under varying but fixed noise power Np.

6.4.1 Algorithm

The algorithm to compute the near-to-optimal transmit power level needs two important considerations. First, it has to be very simple in calculation so that the computation cost remains minimum and importantly the algorithm computation should be fast enough so that the transmit power can be tuned to the best level quickly. Second, it is important to assume a slow varying channel model because even if the best transmit power level is obtained through the transmission and reception of only one packet, it is only worth-full if we consider the same channel conditions for at least 2 to 3 packets. It is important to mention here that adapting the transmit power according to the channel variations does not mean estimating or equalizing the channel but it is just to adapt the power level according to the channel behavior.

The computation of the adaptive transmit power algorithm consists of the following steps and can also be seen in Fig. 6.6.

- As the data packet is received, the received power is calculated from the path losses and the known transmit power level of the transmitter that was updated by the receive node in the last communication. Further received SNR is calculated

by assuming that the noise power is know at the receiver. This assumption is considered during simulation, however, in real life case noise power can be estimated through RSSI (as explained in Sec. 6.6).

- After an SNR estimation, BER is computed (from Eq. 3.26), and minimum threshold 10^{-3} is considered (by using BPSK modulation scheme) as acceptable error rate. After words, the energy per bit is calculated as explained in Sec. 6.3.2.
- For all E_b values against transmit power levels and corresponding possible SNR variations (under fixed noise at a time), there is always a transition point from where the energy per bit starts increasing. This transition point is considered as near-to-optimal point (as it is the minimum of transmit power versus energy consumption) and is identified through the sign of derivative which changes from negative to positive.

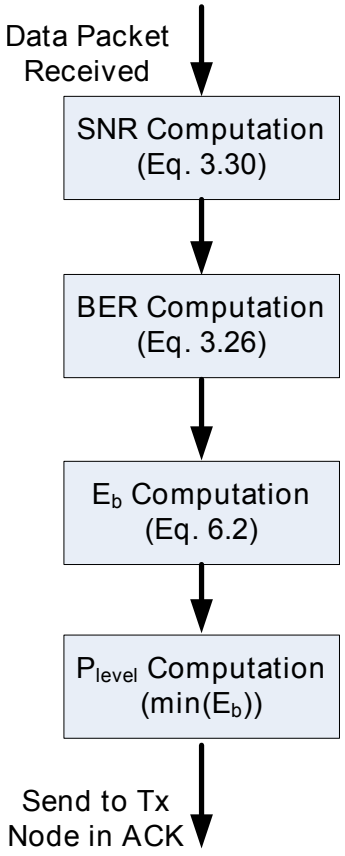


FIGURE 6.6: Execution flow of adaptive transmit power control technique.

The receive node informs the transmit node (using the acknowledgment packet) about the best power level that the transmit node should use to transmit the next data packet under same channel conditions. It is important to mention here that in real-life implementation

of above algorithm a look-up-table (LUT) can be used as an optimization to reduce the processing time. Such an optimization only requires SNR estimation and as a result a corresponding power-level can be selected.

In the next sections, simulation results and performance evaluation are presented by applying the adaptive transmit power technique under two channel profiles.

6.4.2 Simulation Results

Fig. 6.7 shows the energy per successful transmit bit against a huge number of transmit power values. The transmit power has a step of 0.1 over the complete range (i.e. from −20 to 23). The multiple curves shown in Fig. 6.7 correspond to noise power variations of *model-c* which is used to observe the behavior of the adaptive transmit power algorithm under randomly varying SNR. Each curve has an optimal point which is identified and shown in 'black-color'.

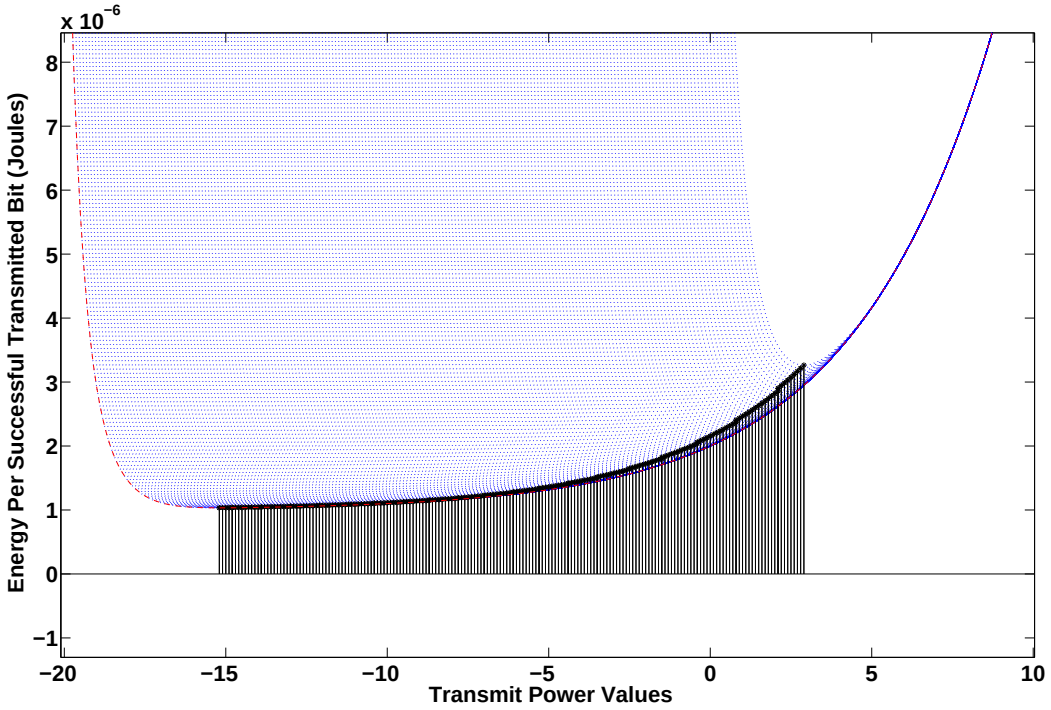


FIGURE 6.7: Energy consumption versus number of different transmit power values. Optimal points on each curve is identified and can be seen by a dark black line. Whereas, the outlined curve (in red) represents the best posssible result for energy efficiency.

Fig. 6.8 shows the best possible energy efficiency at different transmit power levels of *cc-2420* after applying the adaptive transmit power optimization technique under the

channel variations of *model-c* shown in Fig. 6.4-c. The radio transceiver *cc-2420* has 8 different power levels i.e., $0, -1, -3, -5, -7, -10, -15, -25$ dBm and therefore the optimal points have to be shifted to the closest power level in real life implementation. It is found that for high SNR values, lower power levels contain the best point, whereas for low SNR the best point tends to move closer towards the maximum power level and eventually there is no gain due to adaptive transmit power optimization at 0dBm.

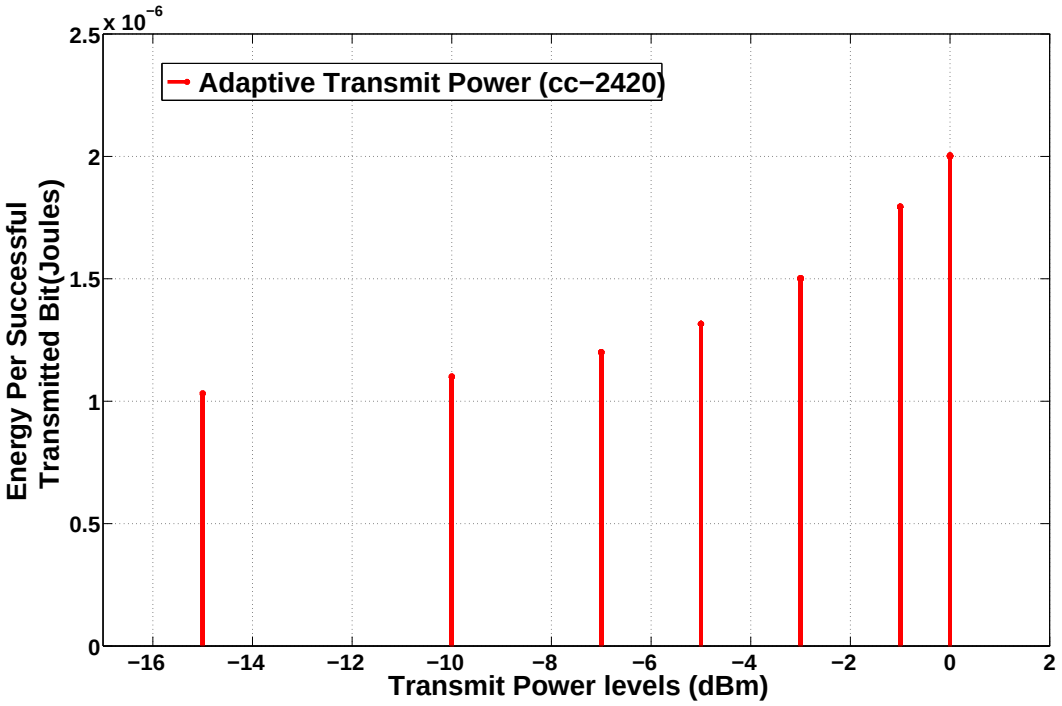


FIGURE 6.8: Best possible energy efficiency versus transmit power levels of *cc-2420* after applying the adaptive transmit power optimization under the channel variations of *model-c*.

Fig. 6.9-b shows the energy consumption for three cases over a transmission period under channel variations from *model-c*. The three cases are fixed transmit power at 0dBm, adaptive (theoretical) transmit power and adaptive transmit power (at available power levels from *cc-2420* radio). It can be seen that the adaptive transmit power technique is always more energy efficient than fixed transmit power (as long as the channel noise is low) and it keeps the track of SNR variation very efficiently. Further, when the channel noise is maximum as it can be seen through several peaks (between 600 – 800s and 1600 – 1800s) in Fig. 6.9-b, there is no significant gain. Moreover, the adaptive transmit power is also tuned to the maximum power level under low SNR (that is energy efficient as low-power levels will result in more energy consumption due to re-transmissions as explained in Sec. 6.3.2).

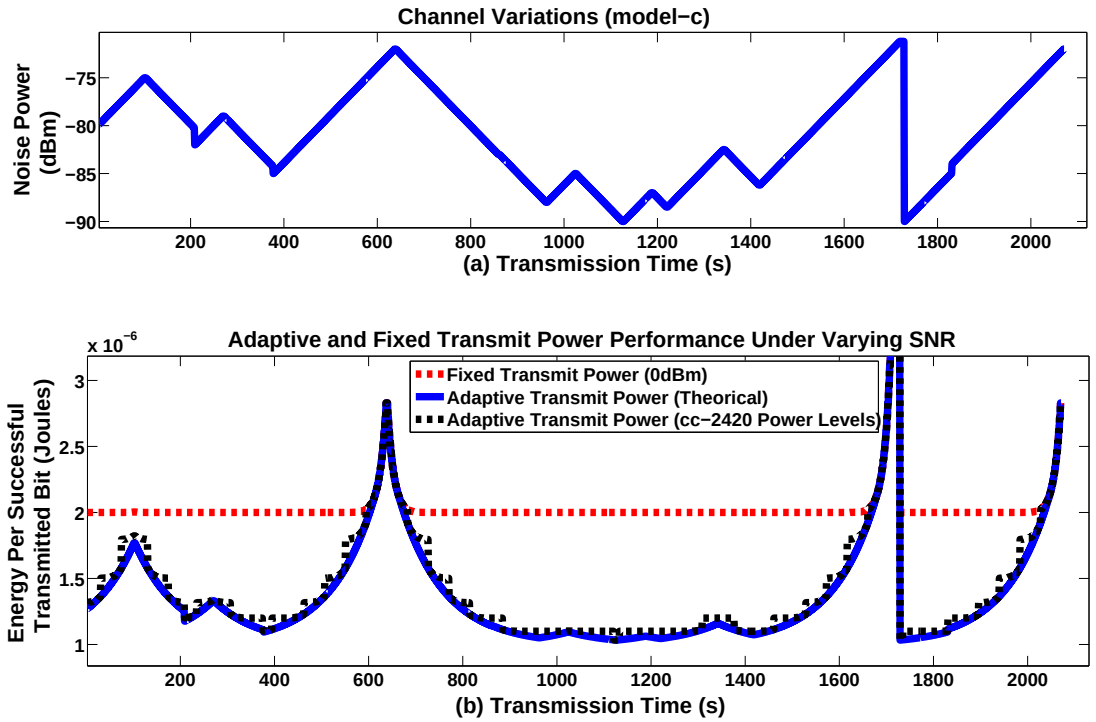


FIGURE 6.9: (a): Channel variations 'model-c'. (b): Performance of the adaptive transmit power technique against the channel variations of *model-c* over time.

Fig. 6.10 shows the best possible energy efficiency versus transmit power levels of *nRF-900* after applying the adaptive transmit power optimization algorithm under the channel variations of *model-d*. The radio transceiver *nRF-900* has only 4 different power levels ($[10, 6, -2, -10]$ dBm) which are much less than *cc-2420*. As the difference between each step is relatively higher, there is a slight reduction of the gain by not having more power levels (closer to theoretical optimal power level).

Fig. 6.11 shows the performance of the adaptive transmit power algorithm against the channel variations of *model-d* over time. The adaptive technique has a significant gain as the power profile of *nRF-900* has a higher slope which helps the adaptive technique to have a better gain over fixed transmit power. It is important to mention that radios operating at sub-giga spectrum usually can transmit the data for a longer distance but when such radio is used for short range transmission we have an average gain in the order of 5 to 8 times in comparison with a fixed transmit power.

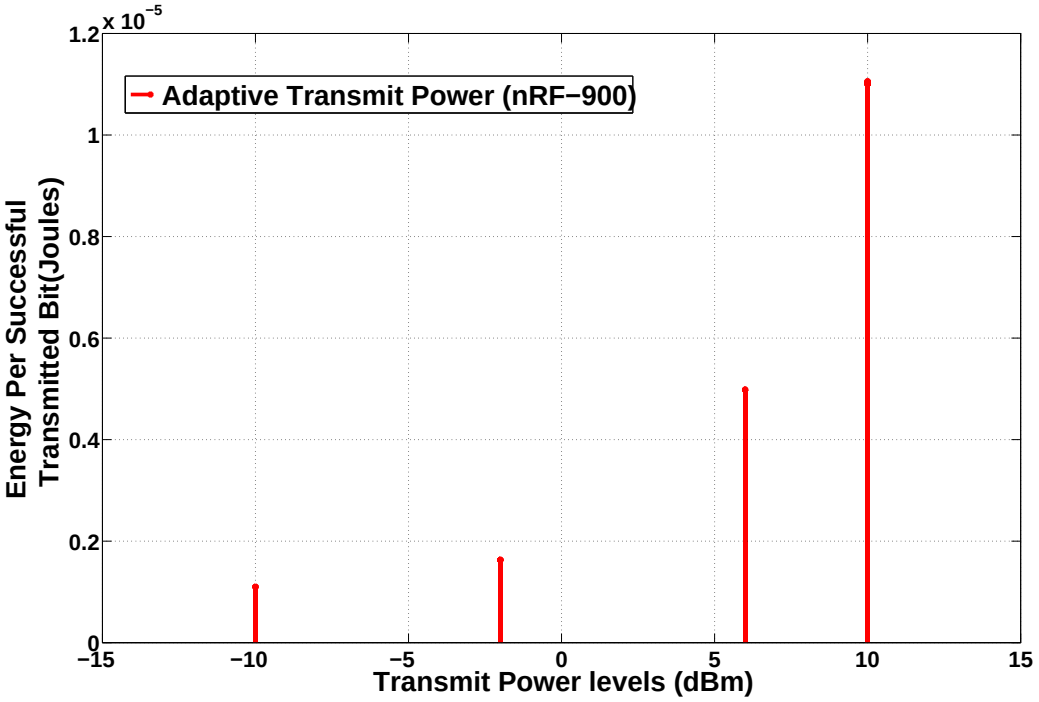


FIGURE 6.10: Best possible energy efficiency versus transmit power levels of *nRF-900* after applying the adaptive transmit power optimization algorithm under the channel variations of *model-d*.

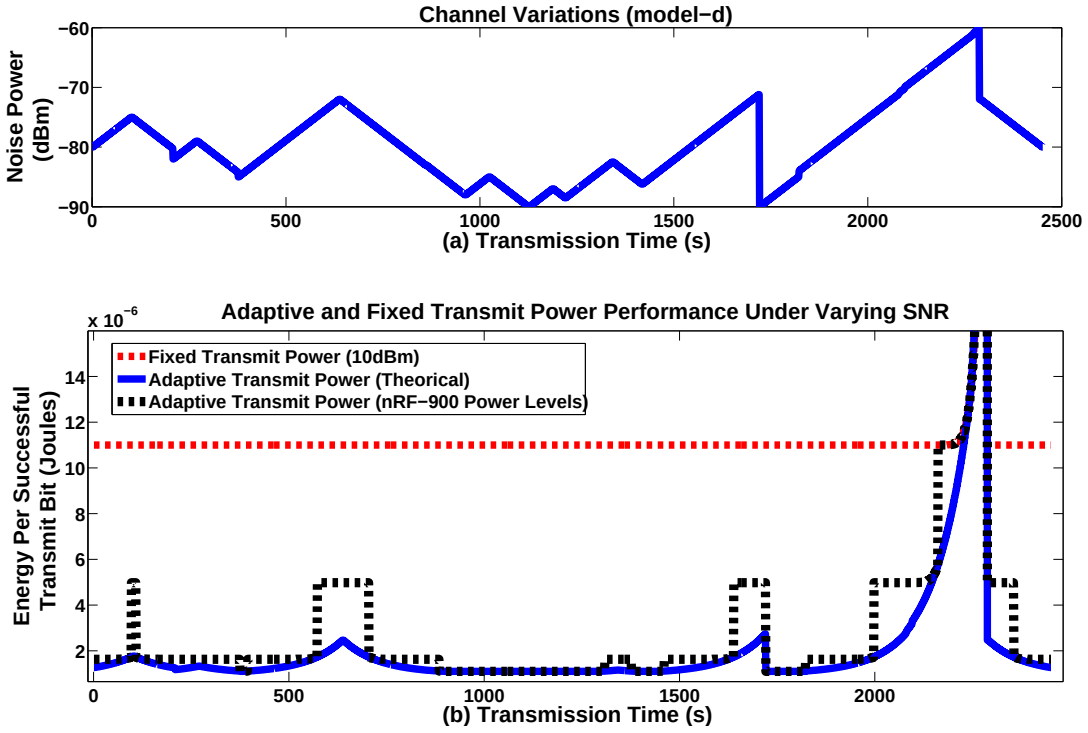


FIGURE 6.11: (a): Channel variations 'model-d'. (b): Performance of adaptive transmit power algorithm against the channel variations of *model-d* over time.

6.5 Performance Evaluation

It is shown in previous section that an adaptive transmit power optimization technique can help to adapt the transmit power to the best possible level accordingly to the channel variations. In order to show the improvements of adaptive transmit power optimizations effectively, it is important to make a quantitative comparison with the static transmit power. In this section the algorithmic proof-of-concept will be extended into performance evaluation results and a comparison with static transmit power consumption. In most of 802.15.4 communications the transmit power is kept fixed to maximum power level. Therefore, 0dBm and 10dBm are the power levels selected from *cc-2420* and *nRF-900* respectively for the comparison.

Fig. 6.12 shows the performance of adaptive transmit power optimization versus fixed transmit power (0 dBm). The *model-c* is applied to evaluate the energy consumption comparison by using *cc-2420* radio chip. It can be seen from Fig. 6.12 that under high and medium SNR values (i.e. for low noise levels), the adaptive transmit power technique is much more efficient and effective, whereas, under low SNR it is slightly better than fixed transmit power level. The overall average energy efficiency gain of adaptive transmit power is between 1.5 to 2 times. When the noise level is too high then the transmit power is automatically set to 0dBm, which means no gain in comparison with fixed power level. It is noteworthy to mention that for very low SNR values where a 0dBm power level is not enough, both adaptive and non-adaptive techniques suffer the same in terms of energy efficiency. Therefore, it was first important to identify the minimum operating SNR for communication at maximum power level. We have to mention that it is not the limit of the adaptive transmit power technique, but rather it is the limit due to low power radio transceiver.

Similarly, Fig. 6.13 shows the performance of adaptive transmit power optimization versus fixed transmit power (10dBm). The *model-d* is used for the comparison of *nRF-900*. It can be seen from Fig. 6.13 as well that under high and medium SNR values the adaptive transmit power technique is much more efficient and effective, whereas, under low SNR it is slightly better than fixed transmit power level. The average gain is between 5 to 6 times, which is much higher in comparison with *cc-2420* because of the power profile characteristics which have significant (current) variations at different levels in comparison with *cc-2420*. On the other hand *nRF-900* transceiver has only four

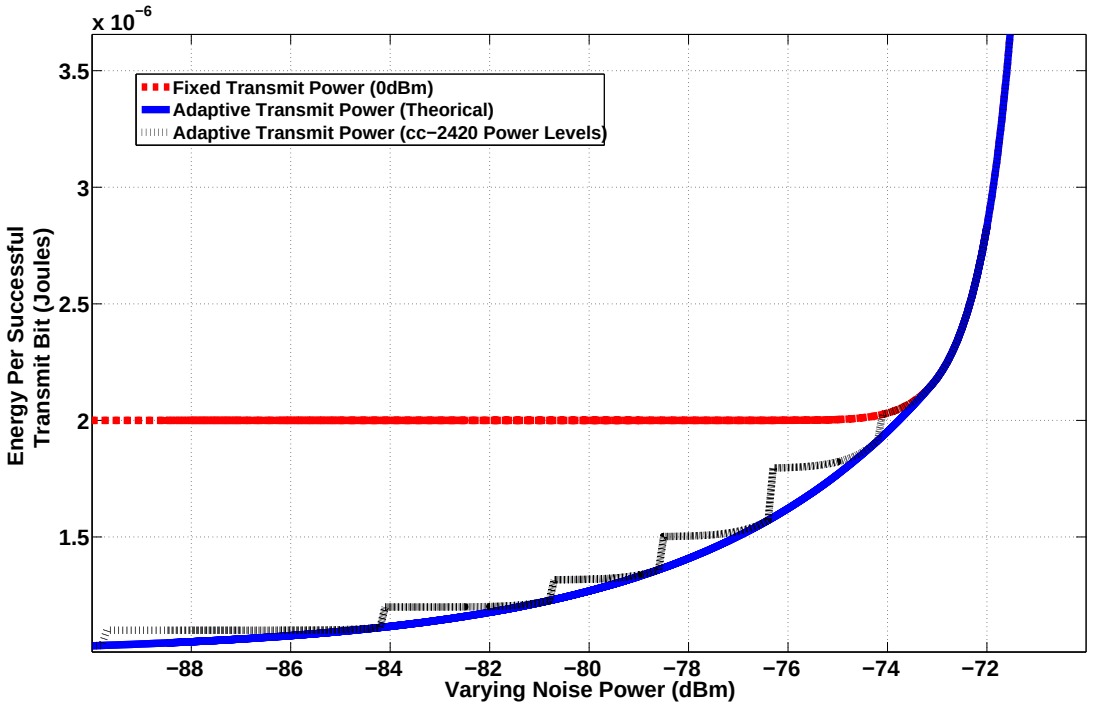


FIGURE 6.12: Comparison of energy performance between an adaptive transmit power optimization versus the fixed transmit power at maximum level of *cc-2420* i.e. (0 dBm).

power levels and hence selecting one of these power levels minimizes the advantage of adaptive transmit power by compromising the optimal power level.

Tab. 6.1 shows the energy consumption of different transmit power variants by using *cc-2420* radio transceiver under high, medium and low SNR which are obtained from *model-c*. It can be observed that the low SNR value matches in both cases because the best adaptive transmit power overlaps with the fixed transmit power at low SNR. The results shown in Tab. 6.1 are for the transmission of single data packet (i.e. 128 bits) after the transmit node has already been tuned to the best power level. It can be seen that by using adaptive transmit power technique the improvements on average are more than 30% which is a significant optimization for just the transmission of two packet under same channel conditions (which means if the transmission rate is increased and more than two packets are transmitted under same channel conditions then the gain can be increased)

Tab. 6.2 shows the energy consumption for a transmission of 1 hr with packet rate of 1 packet/s. This analysis also includes the retransmissions at medium and low SNR as there are no retransmission at high SNR. The proportion of improvements are the same as shown in Tab. 6.1. It is important to mention that the retransmission of packets

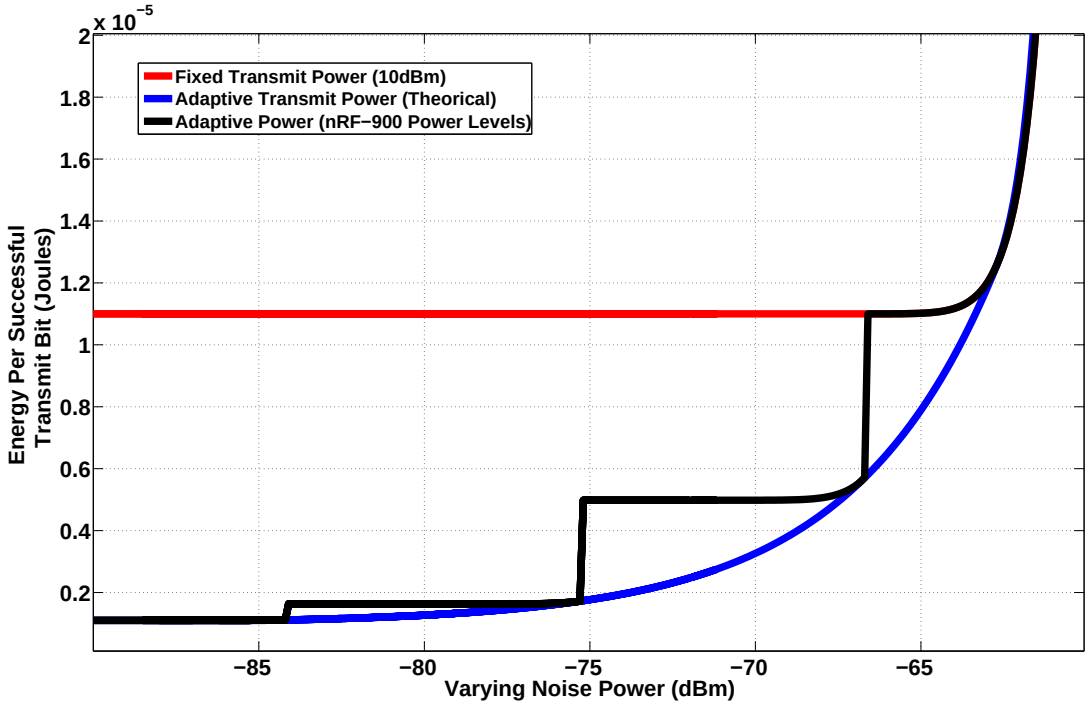


FIGURE 6.13: Comparison between an adaptive transmit power optimization versus the fixed transmit power at maximum level of *nRF-900* i.e. (10dBm).

TABLE 6.1: Performance evaluation of adaptive versus fixed transmit power under channel variations of *channel-c* for *cc-2420* radio transceiver.

Transmit Power	**High SNR (Joules/Packet)**	**Medium SNR (Joules/Packet)**	**Low SNR (Joules/Packet)**
Max. Fixed (0 dBm) (Simulated)	$2.56.10^{-4}$	$2.60.10^{-4}$	$6.08.10^{-4}$
Adaptive (Simulated)	$1.32.10^{-4}$	$1.72.10^{-4}$	$6.08.10^{-4}$
Improvements(%)	49	34	0

are calculated through automatic repeat request (*ARQ*) mechanism as explained in Sec. 3.2.1 (Eq. 3.1), and it is included in the results shown in Tab. 6.2.

Tab. 6.3 shows the energy consumption of different transmit power variants by using *nRF-900* radio transceiver under high, medium and low SNR from *model-d*. Fixed transmit power at 10 *dBm* is used to compare with the adaptive transmit power. Similar to Tab. 6.1, all the results and improvements have been shown for a transmission of one data packet in Tab. 6.3. It can be seen that by using adaptive transmit power technique for *nRF-900* radio transceiver the improvements on average is 86% which is better than that of *cc-2420* because *nRF-900* maximum power is 10dBm greater than *cc-2420*.

TABLE 6.2: Comparison after 1 *hr.* of transmission with packet rate of 1 packet/s under channel variations of *channel-c* for *cc-2420* radio transceiver.

Transmit Power	**High SNR (Joules)**	**Medium SNR (Joules)**	**Low SNR (Joules)**
Max. Fixed (0 dBm) (Simulated)	0.90	0.93	2.18
Adaptive (Simulated)	0.47	0.61	2.18
Improvements(%)	47.8	35	0

TABLE 6.3: Performance evaluation of adaptive versus fixed transmit power under channel variations of *channel-d* for *nRF-900* radio transceiver.

Transmit Power	**High SNR (Joules/Packet)**	**Medium SNR (Joules/Packet)**	**Low SNR (Joules/Packet)**
Max. Fixed (10 dBm) (Simulated)	$1.1.10^{-5}$	$1.1.10^{-5}$	$1.01.10^{-4}$
Adaptive (Simulated)	$1.1.10^{-6}$	$1.63.10^{-6}$	$1.01.10^{-4}$
Improvements(%)	90	86	0

Tab. 6.4 shows the energy consumption for a transmission of 1 *hr* with packet rate of 1 packet/s. This analysis also includes the retransmissions at medium and low SNR as there are no retransmission at high SNR for the case of *channel-d* for *nRF-900* as well. The proportion of improvements on average is 85.4% which is nearly the same as shown in Tab. 6.3 for transmission of 1 packet.

TABLE 6.4: Comparison after 1 *hr.* of transmission with packet rate of 1 packet/s under channel variations of *channel-d* for *nRF-900* radio transceiver.

Transmit Power	**High SNR (Joules)**	**Medium SNR (Joules)**	**Low SNR (Joules)**
Max. Fixed (0 dBm) (Simulated)	0.0396	0.0396	0.3636
Adaptive (Simulated)	0.00396	0.0058	0.3636
Improvements(%)	90	85.4	0

6.6 Implementation Considerations

The adaptive transmit power technique presented shows the proof-of-concept of dynamic adaptation of transmit power under varying channel conditions. The presented approach relies on the SNR as a metric, which is calculated by exactly knowing the noise power

at the receiver end. However, in practice it is difficult to estimate the exact noise power. Though, there are some referred works exist in the literature which evaluate link quality by using SNR metric as explained in Sec. 6.2.1.1.

For the implementation purpose, we can use hardware parameter such as RSSI (received signal strength indicator) to estimate SNR value. RSSI value has a relation with the received input power as explained in Sec. 23 and shown in Fig. 27 of the data sheet of *cc*−2420 radio transceiver. For SNR estimation, the received power can be evaluated during the time when the data packet is received, whereas, immediately after the reception of data packet a control packet can be transmitted including the RSSI value. The received power obtained from this RSSI value can reflect a channel noise power and consequently SNR can be estimated. This estimation process has to be verified in future by conducting some real-time experiments.

As far as the implementation of the adaptive transmit power technique is concerned, the execution flow presented in Sec. 6.4.1 (and shown in Fig. 6.6) can be optimized by using a look-up-table (LUT). After estimating SNR as explained above, an off-line computation of the best power level can be obtained with respect to various SNR values. Tab. 6.5 shows a range of possible SNR values from maximum (i.e. good channel conditions) to minimum (i.e. bad channel conditions) and further worst SNR (i.e. lower than minimum). It can be seen that according to the SNR variations the corresponding power level can be selected using LUT. Therefore, in practice as soon as the SNR value is obtained the appropriate power level can be transmitted in the acknowledgment (from receive node to transmit node) such that the transmit node can use this optimized power level for next transmission. Moreover, the processing of the computation of bit-error-rate and energy-per-bit can be avoided by using LUT based optimization.

SNR Values (dB)	Transmit Power (Plevel)
Max.	Min
-	-
-	-
-	-
-	-
Min.	Max.
Lower than Min.	Max.

TABLE 6.5: Look-up-table for an optimized implementation of adaptive transmit power technique.

6.7 Conclusion

In this chapter an adaptive transmit power control technique is proposed with an objective to optimize the transmit power dynamically in WSN. The main idea behind the proposed approach is to adapt the transmit power according to the signal-to-noise (SNR) variations. A receive node after receiving a data packet calculates the energy consumption per-bit through bit-error-rate, and the energy per-bit is calculated against number of different transmit power values. Then the best transmit power is obtained through a gradient of the curve which identifies the sign of the slope and provide minimum energy per successfully transmitted bit. In this regard, two random noisy channel models are used to evaluate the performance of this adaptive technique. Different radio transceivers were considered at 2.4GHz (*cc-2420* from TI) and 900MHz (*nRF-900* from Nordic Semiconductor) because of their different power profile characteristics. Further the results of energy consumption are computed for single packet transmission under low, medium and high SNR values. Finally the improvements are shown through a comparison of static transmit power with a dynamic power optimizations. It is shown that by adapting the transmit power according to channel variations can improve the energy efficiency by a factor of 2 to 5 times for different radios in comparison with worst-case static power utilization. It is important to mention here that by utilization error correcting codes (ECC) the transmit power can be further improved by 3 to 5 dBm which as a result can be more energy efficient [25].

Conclusion

In the last decade, Wireless Sensor Networks (WSNs) have progressively changed the way to monitor environmental and physical phenomenons. The deployment of tiny embedded devices, capable of operating unattended for long periods of time, has not only transformed communication systems but also added a new dimension in wireless communications. Though, the energy constraint inherent to battery-powered sensor nodes has opened challenges on many different fronts. From the hardware point of view, the radio transceiver is more and more efficient in terms of power consumption and more innovative technological improvements are under way. From the software point of view, protocol stack optimizations not only within each layer but also through cross-layer techniques have achieved significant improvements for extending lifetime of the sensor networks but has not yet reached to the limits. Moreover, as the impact of software controlling the hardware is extremely vital, there is still room for WSN to perform better.

In this thesis software-based algorithmic level power optimization techniques are proposed to not only improve the software energy efficiency but also to optimize those factors that control the hardware and consequently make the hardware utilization more energy efficient. The focus has been on the lower layers of the protocol stack as their contribution towards energy consumption is more dominant. Before to apply innovating yet simple techniques, it is important to identify which optimization could be more significant in overall energy efficiency. Therefore, this thesis has been divided into three parts i.e. energy modeling, MAC-layer and physical-layer optimizations. It is the first part (energy modeling) that has open-up our vision towards exact optimizations for the MAC and physical layers because it has provided us detailed insight of the energy consumption.

In the first contribution a scenario based hybrid energy (MAC) model is presented. The

model is based on analytical analysis and real-time power measurements from hardware/software platform (PowWow). The model helps to evaluate an accurate estimate of energy consumption during the communication between sensor nodes. The model is accurate because of two main reasons. First, it identifies and isolates different traces of scenarios that occur during communication in distributed asynchronous sensor networks. These scenarios provide the current variations within each state which provides more accuracy in comparison to fixed current level states models. Scenarios such as CBT (calculation before transmission), channel sensing, low power modes provide the details of the variable power consumption instead of the states. Second, as the energy consumption of the scenarios is calculated through power measurements it is accurate and therefore we have very low relative error after validation (between 1% *to* 8%).

The second contribution is based on the design of a traffic-aware dynamic MAC protocol with an objective to decrease the energy waste in distributed asynchronous sensor networks such as overhearing, collisions, overheads and most importantly idle listening. First, the TAD-MAC protocol is characterized through parameters of the adaptive algorithm in a network simulator (WSNet). The behavior of the system is modeled as a control system in open-loop and closed-loop forms to understand the convergence behavior of the algorithm. The system is closely related with non-linear autoregressive exogenous time series model, but the formal mathematical proof of the model (which is based on the adaptive algorithm) is highly complex, therefore heuristic-based method is targeted. This approach has helped to simplify the adaptive algorithm complexity by extensive experiments which were performed to ensure the convergence of TAD-MAC and also to determine the performance metrics such as energy consumption, latency, and convergence time. Second, TAD-MAC protocol has been applied in the context of wireless body area sensor networks. TAD-MAC protocol takes into account the traffic variability within each node as well as within the network. The algorithm designed for TAD-MAC not only optimizes energy consumption but it also reduces significantly the delay for packet transmissions. It was shown that it outperforms all the other MAC protocols in the category of preamble sampling MAC protocols and improves the lifetime of typical body area networks by 3 to 6 times in comparison with WiseMAC which is considered as the most efficient preamble sampling MAC protocol.

The third contribution of this thesis is focused on the output transmit power reduction. An adaptive transmit power optimization technique is applied under dynamic channel

variations to reduce the energy per successfully transmitted bit at the physical layer. The output power is adaptively tuned to the best power level on link-by-link basis. Each node locally adapts the power according to SNR variations. The optimization is achieved under slow varying channels at the reception of a single packet. Different radio-transceiver power profiles are used to show the gain over fixed transmit power. It is found that the dynamic adaptation of the transmit power can help to reduce the energy consumption by a factor of 2. Moreover this factor can be further improved if error correcting codes are utilized.

Finally it is important to mention the limitations of the adaptive techniques being applied at the MAC and physical layers. For TAD-MAC it is important that the traffic variations should be slower than the convergence time. If the packet-rate changes before a node has reached a steady state and it is always changing very fast, the adaptive algorithm proposed in TAD-MAC will not be effective, and it will be even more energy consuming than other low power MAC protocols such as wiseMAC. For the adaptive transmit power technique, it is also necessary that at least two packets have same channel conditions, if the channel is varying very fast (less than the time duration of two packets), performance will be degraded.

Several future works can be considered to enhance our work. These can be envisioned first as a global improvement and then within the proposed contributions. From the global WSN point of view, this work could be extended towards global self-organized WSN. This thesis has achieved local-level self-organization which has improved the energy efficiency of the low layers of the protocol stack. To have network-level self-organized WSN, certain rules can be designed to integrate locally optimized parameters and transform them into a global optimization [84] .

The hybrid energy model presented in this thesis is an accurate energy consumption model. Though, certain improvements can be added in it. For example, some parameters such as collision probabilities are already available, but interference probabilities and link quality information can further enhance the accuracy. Furthermore, some specific patterns or features of the data collisions would be interesting to identify.

For MAC layer optimizations, TAD-MAC has unique characteristics and it can adapt quickly to different types of traffic such as burst, periodic and non-periodic. It is important to implement TAD MAC protocol on real WSN motes and further extend it into

multi-hop environment. Also, the mathematical proof of convergence can be considered as future work.

For the case of transmit power optimization, in most of the radio transceivers there are some parameters such as LQI (link quality indicator) and PRR (packet reception ratio). These parameters need to be used as well to adapt the transmit power in real condition. It will be interesting to compare those results with the results presented in Chapter 6. Transmit power can be further reduced if proper error correcting codes (ECC) are utilized [25].

Personal Publications

Journal Articles

- **M. M. Alam**, O. Berder, D. Menard, T. Anger and O. Sentieys, Hybrid Model for Accurate Energy Analysis for WSN nodes, *EURASIP Journal for Embedded Systems, vol. 2011, pp. 4.1-4.16.*
- **M. M. Alam**, O. Berder, D. Menard and O. Sentieys, TAD-MAC: Traffic-Aware Dynamic MAC Protocol for Wireless Body Area Sensor Networks, *IEEE Journal of Emerging And Selected Topics in Circuits and Systems, vol. 2 issue 1, pp. 109-119, March. 2012.*

Conference Articles

- **M. M. Alam**, O. Berder, D. Menard and O. Sentieys, On the Energy Savings of Adaptive Transmit Power for Wireless Sensor Networks Radio Transceivers, *ACM proceedings of 26th International Conference on Architecture of Computing Systems, pp. 6, Feb. 2013, Prague, Czech Republic.*
- **M. M. Alam**, O. Berder, D. Menard and O. Sentieys, Latency-Energy Optimized MAC Protocol for Wireless Body Area Sensor Networks, *IEEE Proceedings of 9th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 67-72, May 2012, London, UK.*
- **M. M. Alam**, O. Berder, D. Menard and O. Sentieys, Traffic-Aware Adaptive Wake-Up-Interval for Preamble Sampling MAC Protocols of WSN, *IEEE 3rd International Workshop on Cross-Layer Design (IWCLD), pp. 1-5, November 2011, Rennes, France.*

- **M. M. Alam**, O. Berder, D. Menard and O. Sentieys, Accurate Energy Consumption Evaluation of Preamble Sampling MAC Protocols for WSN, *ACM proceedings of 24th International Conference on Architecture of Computing Systems, pp. 6, Feb. 2011, Como, Itly.*

Bibliography

[1] *Crossbow.* http://xcrossbow.com.

[2] *IEEE 802.15.4. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR WPANs).*

[3] *Mobility framework.* http://mobility-fw.sourceforge.net/.

[4] *NS2 Simulator.* http://www.isi.edu/nsnam/ns/.

[5] *Power aware wireless sensor networks framework.* http://pawis.sourceforge.net/.

[6] *SENSE: Wireless sensor simulator.* http://www.ita.cs.rpi.edu/sense/index.html/.

[7] Actel Corporation. IGLOO Handbook, 2008.

[8] S. Agarwal, R. Katz, S.V. Krishnamurthy, and S.K. Dao. Distributed power control in adhoc wireless networks. pages 59–66. IEEE Proc. of Int. Symp. Personal, Indoor, and Mobile Radio Comm. (PIMRC), 2001.

[9] A. Ahmad, A. Riedl, W. J. Naramore, N. Y. Chou, and M. S. Alley. Scenarios based traffic modeling for data emanating from medical instruments in clinical environment. pages 529–533. in Proc. of the World Congress on Computer Sci. and Info. Eng., 2009.

[10] M. R. Ahmad, E. Dutkiewicz, and X. Huang. A survey of low duty cycle mac protocols in wireless sensor networks. *InTech Emerging Communications for Wireless Sensor Networks*, pages 69–90, 2011.

[11] I. F. Akyildiz and E. P. Stuntebeck. Wireless underground sensor networks. *J. of Ad Hoc Networks*, 6(6):669–686, 2006.

[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : A survey. *Computer Network Journal*, 38(4):393–422, 2002.

[13] I. F. Akyildiz and M. C. Vuran. *Wireless Sensor Netorks*. John Wiley Publishing Company, 2010.

[14] M. M. Alam, O. Berder, D. Menard, T. Anger, and O. Sentieys. A hybrid model for accurate energy analysis for wsn nodes. *EURASIP Journal for Embedded Systems*, 2011:4.1–4.16, 2011.

[15] M. M. Alam, O. Berder, D. Menard, and O. Sentieys. TAD-MAC: Traffic aware dynamic mac protocol for wireless body area sensor networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 43(1):109–119, 2012.

[16] M.M. Alam, O. Berder, D. Menard, and O. Sentieys. Traffic-aware adaptive wake-up-interval for preamble sampling mac protocols of WSN. In *Third International Workshop on Cross Layer Design (IWCLD)*, pages 1–5, November 2011.

[17] Inc. AMI Semiconductor. *AMIS 52100 Low Power Transceiver with Clock Data Recovery: AMIS 5210 Manual*. http://www.datasheetarchive.com/AMIS-52100*-datasheet.html.

[18] S. Arnon, D. Bhastekar, D. Kedar, and A. Tauber. A comparative study of wireless communication network configurations for medical applications. *IEEE Wireless Communications*, 10(1):56–61, 2003.

[19] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. MAC essentials for wireless sensor networks. *IEEE Communication surveys & Tutorials*, 12(2):222–248, 2010.

[20] N. Bambos and S. Kandukuri. Optimal transmit power in wireless sensor networks. *IEEE Wireless Comm. Magazine*, 9(3):58–64, 2002.

[21] A. Barberis, L. Barboni, and M. Valle. Evaluating energy consumption in wireless sensor networks applications. pages 455–462. Proceedings of the 10th EUROMICRO, 2007.

[22] L. Barboni and M. Valle. Experimental analysis of wireless sensor nodes current consumption. pages 401–406. Second International Conference on Sensor Technologies and Applications (SENSORCOMM), 2008.

[23] O. Berder and O. Sentieys. PowWow: Power optimized hardware/software framework for wireless motes. pages 229–233. Proc. of the Workshop on Ultra Low Power Sensor Networks, 2010.

[24] M. Buettner, E. Yee, Gary V. Anderson, and R. Han. XMAC: A short preamble mac protocol for duty cycled wireless sensor networks. pages 307–320. 4th ACM Conf. on Embedded Networked Sensor Systems (SenSys), 2006.

[25] Mickeal Cartron. *Towards a platform for energy efficient WSN.* PhD thesis, Department of Electrical Engineering and Computer Sciences. University of Rennes1. France., 2006.

[26] INRIA Research Center. *WSIM Simulator.* http://wsim.gforge.inria.fr/download.html.

[27] INRIA Research Center. *WSNet Simulator.* http://wsnet.gforge.inria.fr/download.html.

[28] INRIA Research Center. *WSNet Simulator.* http://ensiwiki.ensimag.fr/index.php/Module_Creation_for_WSNet.

[29] A. Chehri, P. Fortier, and P. M. Tardif. Uwb based sensor networks for localization in minning environments. *J. of Ad Hoc Networks*, 7(5):987–1000, 2009.

[30] Chipcon. *SmartRF CC1000 Single Chip Very Low Power RF Transceiver*, Jan. 2003.

[31] Chipcon. *SmartRF CC2420, 2.4GHz IEEE 802.15.4/ZigBee ready RF Transceiver*, Sep. 2004.

[32] N. Cho, J. Bae, and H. J. Yoo. A 10.8 mw body channel communication/mics dual band transceiver for a unified body sensor network controller. *IEEE J. of Solid State Circuits*, 44(12):3459–3468, 2009.

[33] S. Chouhan, R. Bose, and M. Balakrishnan. A framework for energy-consumption-based design space exploration for wireless sensor nodes. *IEEE Tran. on Computer-Aided Design of Integrated Circuits and Sys.*, 28(7):1017–1024, 2009.

[34] S. Chouhan, R. Bose, and M. Balakrishnan. Integrated energy analysis of error correcting codes and modulation for energy efficient wireless sensor networks. *IEEE Trans. on Wireless Communication*, 8(10):5348–5355, 2009.

[35] A. Courtay, A. Pegatoquet, M. Auguin, and C. Chabaane. Wireless sensor networks node global energy consumption modeling. pages 54–61. Conference on Design and Architectures for Signal and Image Processing (DASIP), 2010.

[36] CrossBow. *Mote Battery Life Calculator PowerManagement.* http://www.xbow.com/.

[37] T. V. Dam and K. Langendoen. An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks. pages 171–180. in Proc. of ACM Conf. on Embedded Network Sensor Systems (SenSys), 2003.

[38] J. D. Decotignie, C.C. Enz, V. Peiris, and M. Hubner. Wisenet: An ultralow power concept for wireless sensor network. *tm - Technisches Messen 2010*, 77:107–112, 2010.

[39] A. Dunkels, B. Gronvall, and T. Voigt. Contiki a lightweight and flexible operating system for tiny networked sensors. pages 455–462. IEEE Proc. First Workshop on Embedded Networked Sensors, 2004.

[40] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He. Software based online energy estimation for sensor nodes. pages 28–32. EMNETS. Proc. 4th workshop on Embedded networked sensors, 2007.

[41] A. Dunkels, O. Schmidt, and T. Voigt. Using protothreads for sensor node programming. Proc. Workshop on Real World Wireless Sensor Networks, 2005.

[42] T. Elbatt, S. Krishnamurthy, D. Connors, and S. Dao. Power management for throughput enhancement in wireless adhoc networks. pages 1506–1513. Proc. IEEE Int. Conf. Comm. (ICC), 2000.

[43] C.C. Enz, A. E. Hoiydi, J. D. Decotignie, and V. Peiris. Wisenet: an ultra low power wireless sensor network solution. *Computer*, 37:62–70, 2004.

[44] C. A. Boano et al. The triangle metric: Fast link quality estimation for mobile wireless sensor networks. pages 1–7. Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN), 2010.

[45] C. M. Rondinone et al. Designing a reliable and stable link quality metric for wireless sensor networks. pages 6–10. Proceedings of the workshop on Real world wireless sensor networks (REALWSN), 2008.

[46] C. W. Mundt et al. A multiparameter wearable physiologic monitoring system for space and terrestrial applications. *IEEE Trans. on Info. Tech. in Biomedicine*, 9(3):382–391, 2005.

[47] C. W. Mundt et al. Monitoring of physiological parameters from multiple patients using wireless sensor network. *J. of Medical Systems*, 32(5):433–441, 2008.

[48] K. S. Kwak et al. *A Traffic based Secure MAC Protocol for WBAN with Bridging Function.* https://mentor.ieee.org/802.15/dcn/09/15-09-0366-00-0006-inha-mac-proposal.pdf.

[49] N. Baccour et al. Flqe: A fuzzy link quality estimator for wireless sensor networks. pages 1–16. The 7th European Conference on Wireless Sensor Networks (EWSN), 2010.

[50] R. Christian et al. Prediction accuracy of link quality estimators. pages 1–16. The 8th European Conference on Wireless Sensor Networks (EWSN), 2011.

[51] et al. N. Baccour. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks*, 8(4), 2012.

[52] T. Falck, H. Baldus, J. Espina, and K. Klabunde. Plug n play simplicity for wireless medical body sensors. *J. of Mob. Netw. Appl*, 12(2-3):143–153, 2007.

[53] S. Farshchi, A. Pesterev, P.H. Nuyujukian, I. Mody, and J. W. Judy. Bi fi: An embedded sensor/system architecture for remote biological monitoring. *IEEE Trans. on Info. Tech. in Biomedicine*, 11(6):611–618, 2007.

[54] Chipcon Products from Texas Instruments. *cc1000 Manual*, 2007. http://focus.ti.com/lit/ds/symlink/cc1000.pdf.

[55] Chipcon Products from Texas Instruments. *cc2420 Manual*, 2007. http://ti.com/lit/ds/symlink/cc2420.pdf.

[56] S. Harte, B. O'Flynn, M. Rafael V, and E. Popovici. Wireless sensor node design for heterogeneous networks. pages 1–7. 32nd International Microelectronics and Packaging IMAPS-CPMT, September 2008.

[57] H. Hashemi. *Book Chapter: Indoor Propagation Modeling.* IEEE Press Series on Digital & Mobile Communication, Wireless Communications in the 21st Century, international edition, 2002.

[58] P. Heinzer, V. Lenders, and F. Legendre. Fast and accurate packet delivery estimation based on dsss chip errors. pages 2916–2920. Proceedings of International Conference on Computer Communications (INFOCOM), 2012.

[59] A. E. Hoiydi and J.D. Decotignie. Wisemac: An ultra low power mac protocol for the downlink of infrastucture wireless sensor network. pages 244–251. In the procedding of 9th IEEE Symposium on Computers and Communications, (ISCC), June 2004.

[60] A. E. Hoiydi, J.D. Decotignie, C. Enz, and E. Le Roux. WiseMac, an ultra low power mac protocol for the wisenet wireless sensor networks. pages 302–303. ACM Conf. on Embedded Networked Sensor Systems (SenSys), 2003.

[61] P. Hurni and T. Braun. MaxMAC: A maximally traffic adaptive mac protocol for wireless sensor networks. pages 289–305. EWSN. 7th European Conference on Wireless Sensor Networks, 2010.

[62] P. Hurni, T. Braun, and M. Anwander. Evaluation of wisemac and extensions on wireless sensor nodes. *J. of Telecom. Systems*, 437(1-2):49–58, 2010.

[63] INRIA. *PowWow Hardware/Software Sensor Platform*, 2010. http://powwow.gforge.inria.fr/.

[64] X. Jiang, P. Dutta, D. Culler, and I. Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. pages 186–195. Proceeding of the Sixth IPSN, 2007.

[65] T.R.F. Jones, W. G. Yeon, and M. Welsh. A portable low power, wireless two lead ekg system. pages 2141–2144. IEEE 26th Annual Inter. Conf. of Eng. in Medicine and Biology Society, 2004.

[66] N. Jongkeun, L. Sangsoon, and K. Chong-Kwon. Dual wakeup low power listening for duty cycled wireless sensor networks. *EURASIP. Wireless Communications and Networking Journal*, 2008(11):46:1–46:11, 2008.

[67] Baoqiang Kan, Li Cai, Lei Zhao, and Yongjun Xu. Energy efficient design of wsn based on an accurate power consumption model. pages 2751–2754. International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2007.

[68] H. Karl and A. Willig. *Protocol and Architecture for Wireless Sensor Networks.* John Wiley & Sons, Ltd., international edition, 2005.

[69] S. Kellner, M. Pink, D. Meier, and E. O. Blass. Towards a realistic energy model for wireless sensor networks. pages 97–100. IEEE Proc. 5th Annual Conference on Wireless On demand Network Systems and Services, 2008.

[70] F. Kerasiotis, A. Prayati, C. Antonopoulos, C. Koulamas, and G. Papadopoulos. Battery lifetime prediction model for a wsn platform. pages 525–530. in Proc. of Fourth Inter. Conf. on Sensor Tech. and Applications, 2010.

[71] Z. A. Khan. *Cross Layer Design for Energy Constrained Wireless Sensor Networks.* PhD thesis, University of Nice Sophia Antipolis, France., 2011.

[72] M. Kubisch, H. Karl, A. Wolisz, L.C. Zhong, and J. Rabaey. Distributed algorithms for transmission power control in wireless sensor networks. pages 16–20. In IEEE WCNC, 2003.

[73] S. Kulkarni and M. Arumugam. SS-TDMA: A self stabilizing mac for sensor networks. In Sensor Network Operations. Wiley IEEE press edition, 2006.

[74] V. Kumar, D. Rus, and S. Singh. Robot and sensor networks for first responders. *J. of IEEE Pervasive Computing*, 3:24–33, 2004.

[75] K. S. Kwak, M. A. Ameen, D. Kwak, C. Lee, and H. Lee. A study on proposed IEEE 802.15.6 WBAN MAC protocols. pages 834–840. 9th Int. Symp. on Comm. and Information Technology (ISCIT), 2009.

[76] O. Landsiedel, K. Wehrle, and S. Gotz. Accurate prediction of power consumption in sensor networks. pages 37–44. IEEE Proc. Workshop on Embedded Networked Sensors, 2005.

[77] P. J. Lee. Computation of the bit error rate of coherent m ary psk with gray code bit mapping. *IEEE Trans. Communication*, 34(5):488–491, 1986.

[78] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM accurate and scalable simulation of entire tinyos applications. pages 126–137. in Proc. of ACM Conf. on Embedded Network Sensor Systems (sensys), 2003.

[79] M. Li and Y. Liu. Underground coal mine monitoring with wireless sensor networks. *ACM Transaction on Sensor Networks*, 5:10:1–10:29, 2009.

[80] E. Y. Lin. *A Comprehensive Study of Power Efficient Rendezvous Schemes for Wireless Sensor Networks*. PhD thesis, Department of Electrical Engineering and Computer Sciences. University of California Berkeley., 2005.

[81] E. Y. Lin, A. Wolisz, and J. Rabaey. Receiver initiated rendezvous schemes for sensor networks. IEEE Proc. in Communication Society (GLOBECOM), 2005.

[82] E. Y. Lin, A. Wolisz, S. Wiethoelter, and J. Rabaey. Power efficient rendez vous schemes for dense wireless sensor networks. pages 3769–3776. IEEE Proc. Int. Conf. on Comm. (ICC), 2004.

[83] M. Lin, Y. Wu, and I. Wassell. Wireless sensor networks: Water distribution monitoring system. pages 775–778. in Proc. IEEE Radio and Wireless Sympossium, 2008.

[84] K. L.Mills. A brief survey of self organization in wireless sensor networks. *ACM Journal on Wirel. Commun. Mob. Comput.*, 7(1530-8669):823–834, 2007.

[85] A. Manjeshwar, F. Herrmann, E. U. Biyikoglu, and A. Keshavarzian. Measurement and characterization of link quality metrics in energy constrained wireless sensor networks. pages 446–452. IEEE Global Telecommunications Conference, (GLOBECOM), 2003.

[86] H. Matti, P. Pekka, S. Zach, and I. Jari. *Wireless Applications in Healthcare and Welfare Advances in Mobile and Wireless Communications*. Springer Berlin Heidelberg, 2008.

[87] G. G. Messier and I. G. Finvers. Traffic models for medical wireless sensor networks. *IEEE Communication Letters*, 11(1):13–15, 2007.

[88] Network and Pervasive Computing Program of National ICT Australia. *CASTALIA Simulator*. http://castalia.npc.nicta.com.au.

[89] University of California LA. *A Simulation Framework for Sensor Networks*, 2010. http://nesl.ee.ucla.edu/projects/sensorsim/.

[90] University of TU Wien. *Global Energy Optimization for Distributed Embedded Systems*, 2011. http://www.http://geodes.ict.tuwien.ac.at.

[91] University of Twente CTIT. *Global Energy Optimization for Distributed Embedded Systems*, 2010. http://www.utwente.nl/ctit/research/projects/concluded/international/other/geodes.doc/.

[92] B. O'Flynn, A. Barroso, S. Bellis, and J. Benson. The development of a novel miniaturised modular platform for wireless sensor networks. pages 370–375. International Conference on Information Processing in Sensor Networks (IPSN), April 2005.

[93] O. Omeni, A. C. W. Wong, A. J. Burdett, and C. Toumazou. Energy efficient medium access protocol for wireless medical body area sensor networks. *IEEE Trans. on Biomed. Circuits and Syst.*, 2(4):251–259, 2008.

[94] S. Panichpapiboon, G. Ferrari, and O. K. Tonguz. A simple distributed autonomous power control algorithm and its convergence. *IEEE Transactions on Mobile Computing*, 5(10):1432–1447, 2006.

[95] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. pages 95–107. 2nd ACM Conf. on Embedded Networked Sensor Systems (SenSys), November 2004.

[96] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: Enabling ultra low power wireless research. pages 364–369. IPSN/SPOTS, 2005.

[97] A. Prayati, Ch. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos. A modeling approach on the telosb wsn platform power consumption. *Journal of Systems and Software*, 83:1355–1363, 2010.

[98] J. G. Proakis. *Digital Communications*. 4th ed., New York: McGraw Hill, 2001.

[99] Q.Wang, M.Hempstead, and W. Yang. A realistic power consumption model for wireless wireless sensor network devices. pages 286–295. 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, (SECON), 2006.

[100] J. M. Rabaey, M. J. Ammer, J. L. d. Silva, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra low power wireless networking. *IEEE Computer Magazine*, 33(7):42–48, 2000.

[101] R. Ramanathan and R. R. Hain. Topology control of multihop wireless networks using transmit power adjustment. pages 404–413. IEEE Proc. of 19th Joint Conf. of the IEEE Computer and Comm. Societies (INFOCOM), 2000.

[102] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentuce Hall, 2002.

[103] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Comm.*, 17(8):1333–1344, 1999.

[104] E. R. Sanchez, C. Chaudet, and B. Montrucchio. An energy consumption model of variable preamble sampling mac protocols for wireless sensor networks. pages 2285–2289. IEEE 20th Int. Symp. On Personal Indoor and Mobile Radio Communications (PIMRC), 2009.

[105] Nordic Semiconductors. *nRf905 Manual*, 2008. http://www.semiconductorstore.com.

[106] O. Sentieys, O. Berder, P. Quemerais, and M. Cartron. Wake-up interval optimization for sensor networks with rendez-vous schemes. In *Workshop on Design and Architectures for Signal and Image Processing (DASIP)*, Grenoble, France, November 2007.

[107] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod. Cross layer design of ad hoc networks for real time video streaming. *IEEE Comm. Mag.*, 12(4):59–65, 2005.

[108] SHIMMER. *Shimmer Wearable Wireless Sensor Platform*, 2008. http://www.shimmer-research.com/.

[109] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welch. Simulating the power consumption of large scale sensor network applications. pages 188–200. Proc. of ACM Conf. on Embedded Networked Sensor Systems (SenSys), 2004.

[110] M. K. Simon and M. S. Alouini. *Digital Communication over Fading Channels A Unified Approach to Performance Analysis.* John Wiley & Sons, Ins., 2000.

[111] B. Sklar. *Digital Communications Fundamentals and Applications.* 2th ed., Printic Hall, Inc., 2001.

[112] S.Marinkovic, E. Popovici, C. Spagnol, S. Faul, and W. Marnane. Energy efficient low duty cycle mac protocol for wireless body area networks. *IEEE Trans. on Info. Tech. in Biomed.*, 13(6):915–925, 2009.

[113] K. Srinivasan and P. Levis. Rssi is under appreciated. pages 1–6. Proceedings of 3rd Workshop on Embedded Networked Sensors, 2006.

[114] Y. Sun, O. Gurewitz, and D.B Johnson. RI-MAC: a receiver initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. pages 1–14. in Proc. of ACM Conf. on Embedded Network Sensor Systems (SenSys), 2006.

[115] R. F. Mc Sweeney. *Energy Efficient Link Layer Coding and Compression for Wireless Body Area Network.* PhD thesis, Dept. of Elect. and Elec. Eng. Univ. College Cork. Ireland., 2011.

[116] Berkeley WEBS (Wireless Embedded Systems). *Autonomous sensing and communication in a cubic millimeter*, 2010. http://robotics.eecs.berkeley.edu/ pister/SmartDust/.

[117] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. EM-MAC: a dynamic multichannel energyefficient mac protocol for wireless sensor networks. pages 23:1–23:11. in Proc. of 12th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), 2011.

[118] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. PW-MAC: An energy efficient predictive wakeup mac protocol for wireless sensor networks. pages 1305–1313. IEEE 31st International Conference on Computer Communications (INFOCOM), 2011.

[119] Texas Instruments. MSP430 User Guide. Tech. Report, 2009.

[120] S. Ullah and K. S. Kwak. Performance study of low power mac protocols for wireless body area networks. pages 112–116. IEEE 21st Int. Symp. on Personal, Indoor and Mobile Radio Communications, (PIMRC), 2010.

[121] Auburn University. *SensIT Project*, 2010. http://www.eng.auburn.edu/users/lim/sensit.html.

[122] S. G. Valenzuela, M. Chen, and V. C. M. Leung. Mobility support for health monitoring at home using wearable sensors. *IEEE Trans. on Info. Tech. in Biomedcine*, 15(4):539–549, 2011.

[123] A. Varga and H. Rudolf. An overview of the omnet++ simulation environment. pages 60:1–60:10. Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, 2008.

[124] M. C. Vuran and I. F. Akyildiz. Error control in wireless sensor networks: A cross layer analysis. *IEEE Trans. on Networking*, 17(4):1186–1199, 2009.

[125] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. pages 1388–1397. IEEE Proc. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2001.

[126] W. Ye, J. Heidemann, and D. Estrin. An energy efficient mac protocol for wireless sensor networks. pages 1567–1576. in Proc. of IEEE Int Conf. on Communications (ICC), 2002.

[127] J. Yoo, L. Yan, S. Lee, Y. Kim, and H. J. Yoo. A 5.2 mw self configured wearable body sensor network controller and a 12 uw wirelessly powered sensor for a continuous health monitoring system. *IEEE J. of Solid State Circuits*, 45(1):178–188, 2010.

[128] M.R. Yuce. Implementation of wireless body area networks for healthcare systems. *Sensors and Actuators*, 162(1):116–129, 2010.

[129] L. Zheng, W. Wang, A. Mathewson, B. O. Flynn, and M. Hayes. An adaptive transmission power control method for wireless sensor networks. Irish Conference on Signal Processing (ISSC), 2010.

[130] T. Zheng, S. Radhakrishnan, and V. Sarangan. PMAC: An adaptive energy efficient mac protocol for wireless sensor networks. pages 1–8. in Proc. of 19th IEEE Int. Parallel and Distributed Processing Symposium (IPDPS), 2005.

[131] L. C. Zhong, J. Rabaey, and A. Wolisz. An integrated data link energy model for wireless sensor networks. pages 3777–3783. IEEE Int. Conf. on Comm. (ICC), 2004.

Abstract:

Wireless Sensor Networks (WSN) are a fast emerging technology with potential applications in various domains of daily-life, such as structural and environmental monitoring, medicine, military surveillance, robotic explorations etc. WSN devices are required to operate for a long time with limited battery capacity, therefore, the most important constraint in WSN is energy consumption. In this thesis, we propose algorithmic-level dynamic and adaptive optimization techniques for energy reduction in WSN. First, an accurate energy model is presented. This model relies on real-time power measurements of various scenarios that can occur during communication between sensor nodes. It is concluded that MAC layer plays a pivotal role for energy reduction. Then, a traffic-aware dynamic MAC protocol is presented which dynamically adapt the wake-up schedule of sensor nodes through traffic estimation. An adaptive algorithm is designed for this purpose that is heuristically modeled to understand the convergence behavior of algorithmic parameters. The proposed protocol is applied to body area networks and it outperforms other low-power MAC protocols in terms of latency as well as energy consumption and consequently increases the lifetime from three to six times. Finally, an SNR-based adaptive transmit power optimization technique is applied under time-varying channels. The output power is dynamically tuned to best power level under slow varying channel, which results in an average gain by two times.

Key words: WSN, Hybrid energy model, Traffic estimation, Adaptive algorithm, Network simulator, BAN, Adaptive transmit power optimization,

Résumé: Les Réseaux de capteurs sans fil (WSN) sont une technologie émergente avec des applications potentielles dans divers domaines de la vie quotidienne, tels que la surveillance structurelle et environnementale, la médecine, la surveillance militaire, les explorations robotisées, etc. Les nœuds de capteurs doivent de fonctionner pendant une longue période avec des batteries capacité limitée, par conséquent le facteur plus important dans les WSN est la consommation d'énergie. Dans cette thèse, nous proposons des techniques d'optimisation algorithmiques dynamiques, et adaptative pour la réduction de l'énergie. Tout d'abord, un modèle énergétique précise est présenté. Ce modèle repose sur des mesures réelles de courant consommé pour différents scénarios qui peuvent se produire lors de la communication entre les. Il en est conclu que la couche MAC joue un rôle essentiel dans la réduction de l'énergie consommée. Ensuite, un protocole MAC dynamique est présenté. Il adapte de manière dynamique intervalle de réveil des nœuds de capteurs à partir d'une estimation du trafic. L'algorithme adaptatif modélisé de façon heuristique pour comprendre le comportement de convergence des paramètres algorithmiques. Le protocole est appliqué sur des réseaux de capteurs corporels et il surclassant les autres protocoles MAC en termes de latence ainsi qui de consommation d'énergie ce qui permet donc permettant d'augmenter la durée de vie de trois à six fois. Enfin, une technique basée sur l'optimisation adaptative de la puissance d'émission radio est appliquée sur des canaux variant dans le temps. La puissance de sortie est réglée dynamiquement au meilleur niveau de puissance selon l'état du canal, ce qui entraîne un gain moyen d'un rapport deux.

Mots clés: WSN, modèle énergétique hybride, estimation de trafic, algorithme adaptatif, simulateur de réseau, BAN, optimisation adaptive de la puissance d'émission,