DOCTORAL DISSERTATION

Medium Access Control Protocols for Energy and Delay Efficient Applications of Wireless Sensor Networks

Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) am Fachbereich Mathematik und Informatik der Freien Universität Berlin

vorgelegt von:
Pardeep Kumar, M.E.

Tag der Einreichung: 11 Januar, 2012
Tag der Disputation: 20 April, 2012

Gutachter: Prof. Dr. Mesut Güneş
Prof. Dr. Neeli R. Prasad

Institute of Computer Science, Freie Universität Berlin

Acknowledgment

It is my pleasure to write down a few lines about the people who made my Ph.D. a nice, interesting, and rewarding experience. I found them always with me whenever I looked around. I would like to thank them for motivating and encouraging me during this venture.

First and foremost, I am proudly grateful to my supervisor, Prof. Dr. Mesut Güneş, whose thoughtful and logical guidance, insightful vision, continuing support, and frank nature have leaded me to grow immensely over the couple of years. He not only motivated me on technical grounds and provided friendly and enlightening environment but also gave me full freedom to explore my research interests. How to start with research, how to formulate research problems, how to narrow down research interests, what is the technical writing and many more 'hows' and 'whats' I have learnt from him. His suggestions other than technical ones are also of higher value to my future life. In simple, this work would have never happened if it wasn't with him.

I would like to express my special gratitude to Prof. Dr. Jochen Schiller for being cooperative and helpful throughout these years. Taking this opportunity, I would also like to thank Dr. Abd-Al-Basset Al-Mamou, Bastian Blywis, Qasim Mushtaq, Felix Juraschek, Hans Peter Heitzman, and all other colleagues for their constant support and help.

I am deeply indebted to my wife and children for bearing with me during my research time and for being away from parents, relatives, and friends whilst sacrificing own education, language, culture, and much more. I have no words to explain incessant support from my mother, brothers, sisters, and friends during this time. This achievement is only possible with their understanding, patience, and encouragement. Thank you very much.

I present my profound thanks to Higher Education Commission of Pakistan (HEC) and Deutscher Akademischer Austausch Dients (DAAD) for their financial support.

Finally, I would like to thank everyone who, knowingly or otherwise, has provided support and assistance all the way.

Dedication

- To my brother, Jai Parkash ‘Babu’, who died suddenly aged only 44 during this thesis write-up, RIP Babu
- To my adorable mother who is the symbol of love, dedication, courage, hard work, and scarification
- To my sensible wife and lovely kids who simply make my life complete

Abstract

The swift evolution of computation and communication technologies has transferred low-power, low-cost, small-size, and self-organizing devices from obscurity into the complete reality via several hardware forms, each targeting specific applications and tasks. Wireless Sensor Networks (WSNs) are one of such emerging technologies being used in a wide spectrum of applications. However, in parallel to these advancements, WSNs pose design challenges due to the limited resources available to the sensor nodes. The most substantial challenge facing these networks is the requirement of significant reductions in energy consumption of the sensor nodes.

A Medium Access Control (MAC) protocol for WSNs greatly influences the energy consumption of sensor nodes by controlling the functionalities of radio, which is the most power consuming component of a sensor node. Several WSN MAC protocols that have been proposed over the last years are, therefore, mainly tailored towards the *single* prime objective of energy efficiency. Due to the continuing proliferation of WSNs in many diverse applications, it becomes increasingly important to their communication protocols, particularly MAC protocols, not to restrain themselves with the energy efficiency objective. Considering the speedy achievements in producing ultra low power microprocessors and transceivers and in energy harvesting technologies, we conceive that in order to efficiently utilize sensor networks on a large scale for a long term and in a wide range of applications, energy efficiency, even though being the most critical metric, is not a sufficient challenge to address.

This thesis mainly contributes towards the design of a novel WSN MAC protocol called AREA-MAC that considers, beside energy consumption, packet delay, packet delivery, redundancy, traffic adaptability, scalability, etc., for sensor nodes. The AREA-MAC protocol significantly reduces energy consumption of nodes through dealing with collision, idle listening, overhearing, over-emitting, and control overhead. At the same time, it substantially reduces packet delay by enabling the receiving node to respond early and adaptively to the sending node. The protocol efficiently reduces packet redundancy and enables nodes in adapting to the network and traffic conditions. The AREA-MAC protocol uses the channel polling scheme to access the channel, replaces long preambles with the short and smart ones, adds vital additional information to each of them, and employs several other novel approaches in order to achieve the required design goals. Moreover, unlike many other WSN MAC protocols, AREA-MAC does not need any sort of synchronization, time/frequency scheduling, and clustering among nodes, which greatly reduces overhead for nodes and helps a WSN to scale rather smoothly.

Results that have been obtained from analytical, simulation, and real testbed experiments confirm the significant improvements of the AREA-MAC protocol over other state of the art protocols for several performance metrics. These results

have been achieved under a wide range of experiments for several duty cycle values and with varying network scale, topology, and traffic conditions. The effect of cross-layering on the AREA-MAC nodes has also been analyzed. By using different optimization problems, the AREA-MAC has also achieved optimal duty cycle values for sensor nodes that could provide minimum energy consumption and minimum packet delay while keeping the packet delivery ratio above a threshold for a WSN. Similarly the evaluation results obtained from the testbed implementation establish a close match between both simulation and testbed platforms that verifies the acceptability and reliability of the used simulation models and parameter configuration.

Additionally, the emerging IEEE 802.15.4 MAC protocol has been critically analyzed and its limitations, particularly related to energy, delay, and bandwidth have been explored. This manuscript also proposes a simple yet effective 802.15.4 BED scheme that looks into some vital enhancements to the 802.15.4 MAC protocol in order to make it compatible for energy, time, and bandwidth critical WSN applications.

Keywords - wireless sensor networks, medium access control protocols, performance evaluation, energy efficiency, packet delay, packet delivery ratio, channel polling, short preambles, asynchronous, traffic adaptability, IEEE 802.15.4

Zusammenfassung

Die rasante Entwicklung der Computer- und Kommunikationstechnologien in den letzten Jahrzenten hat energiesparsame, günstige und sich selbstorganisierende Kleinstrechner und eingebettete Systeme ermöglicht, welche gezielt für bestimmte Anwendungen und Aufgaben eingesetzt werden können und neue Einsatzgebiete hervorgebracht haben. Drahtlose Sensornetze (Wireless Sensor Network, WSN) sind eine der neuen Technologien aus einer großen Palette von neuartigen Anwendungen, die ermöglicht worden sind. Trotz der vielen Verbesserungen, die erzielt werden konnten, gibt es immer noch offene Fragen bezüglich des Designs von WSNs und den spezifischen Protokollen, da den Sensorknoten nur begrenzte Ressourcen zur Verfügung stehen. Eine der wichtigsten Fragen in solchen Netzen ist die Reduzierung des Energiebedarfs.

Optimierungen sind auf allen Schichten des ISO/OSI Referenzmodells erforderlich. Vor allem beeinflusst das Medienzugriffsprotokoll (Medium Access Control, MAC) den Energieverbrauch der Knoten. Es steuert den Zugriff des Transceivers auf das Medium, wobei die Funkkommunikation der größte energieverbrauchende Faktor eines Knotens ist. Verschiedene MAC-Protokolle sind in den letzten Jahren vorgeschlagen und studiert worden, um primär die Energieeffizienz zu verbessern. Da die WSNs verstärkt Anwendung finden, wird es immer wichtiger, dass deren Kommunikationsprotokolle, insbesondere die MAC-Protokolle, sich nicht nur mit Energieeffizienzproblematik befassen. Betrachtet man die rasante Entwicklung bei der Produktion von ultra-energiesparsamen Mikroprozessoren und Transceivern, sowie bei anderen energiesparenden Technologien, so könnte man annehmen, dass die Energieeffizienz nicht mehr die größte Herausforderung darstellt und weitere Anforderungen zunehmend wichtiger werden.

In dieser Arbeit habe ich mich mit dem Design und der Entwicklung eines neuartigen WSN MAC-Protokolls (genannt AREA-MAC) befasst, welches zusätzlich zum Energieverbrauch weitere Parameter wie Verzögerung, Auslieferungsrate, Redundanz, Anpassungsfähigkeit, Skalierbarkeit, u.s.w. umfasst. Das AREA-MAC Protokoll minimiert den Energieverbrauch von Knoten durch die Reduktion von Paket-Kollisionen, Idle Listening, Overhearing, Over-Emitting und Control Overhead. Gleichzeitig reduziert es die Verzögerung erheblich, indem es dem Empfänger erlaubt früher und gezielter auf den Sender zu reagieren. Das Protokoll reduziert die Redundanz und erlaubt den Knoten sich an die Bedingungen im Netzwerk anzupassen. Das AREA-MAC Protokoll benutzt das Channel Polling Schema um den Zugang auf das Medium zu realisieren, nutzt kurze Präambeln mit zusätzlichen Steuerungsinformationen und benutzt eine Vielzahl von weiteren Möglichkeiten, um die notwendigen Anforderungen zu erzielen.

Analytische Ergebnisse, Simulationen und Experimente in einem Testbed bestätigen eine signifikante Verbesserung durch das AREA-MAC Protokoll verglichen mit alternativen Ansätzen. Die gewonnenen Daten zeigen, dass diese Leistungssteigerung

sich in einer Vielzahl von Topologien, mit verschiedenen Datenverkehrsbedingungen, sowie unterschiedlichen Duty Cycle Einstellungen. Dabei wurden auch Anpassungen auf anderen Protokollschichten betrachtet. Durch das Lösung von Optimierungsproblemen, wurde für das AREA-MAC ein optimaler Duty Cycle bestimmt, welcher minimalen Energieverbrauch sowie Verzögerung erlaubt, dennoch bleibt die Auslieferungsrate über einem gewünschten Schwellenwert. Die Analyse der gewonnenen Ergebnisse der Testbed-Implementierung bestätigt eine starke Korrelation zwischen den Simulationen und den Ergebnissen aus der realen Welt. Dies belegt die Aussagekraft der gewählten Simulationsmodellen und Parameterkonfigurationen.

Darüber hinaus wird das IEEE 802.15.4 MAC Protokoll betrachtet, um Leistungsgrenzen bezüglich des Energieverbrauchs, der Verzögerung und der Bandbreite aufzuzeigen. Diese Arbeit schlägt das 802.15.4 BED Schema vor, dass die Leistungsmetriken für kritische WSN Anwendungen verbessert.

Contents

1 Introduction **1**

1.1 Background and Motivation 2

1.2 Thesis Contribution and Organization 4

2 Wireless Sensor Networks **9**

2.1 Architecture 9

2.1.1 Node Architecture 10

2.1.2 Network Architecture 11

2.2 Applications 12

2.3 Design Objectives and Challenges 14

2.4 MAC Basics for WSNs 15

2.4.1 MAC Services 16

2.4.2 MAC Challenges 17

2.4.3 Common MAC Approaches 20

2.5 Conclusion 25

3 Overview of State of the Art **27**

3.1 Channel Accessing Chronology 27

3.2 Classification of WSN MAC Protocols 31

3.2.1 Contention Based MAC Protocols 31

3.2.2 Scheduling Based MAC Protocols 37

3.2.3 Channel Polling Based MAC Protocols 41

3.2.4 Hybrid MAC Protocols 47

3.3 Comparison of Different WSN MAC Protocols 50

3.4 Conclusion 52

4 Evaluation Methods and Metrics **55**

4.1 Evaluation Methods 55

4.1.1 Analytical Method 56

4.1.2 Simulation Method 57

4.1.3 Real Experimentation Method 60

4.2 Performance Metrics 63

4.2.1 WSN Lifetime 63

4.2.2 End-to-End Delay 64
4.2.3 Packet Delivery Ratio 65
4.2.4 Number of Preambles Sent and Received 65
4.2.5 Number of Overheard Packets 65
4.2.6 Scalability 66
4.2.7 Adaptability with Varying Traffic and Topology Conditions 66
4.2.8 Duty Cycle Effect 66
4.2.9 Cross-Layering 66
4.3 Presentation of Results 67

5 The IEEE 802.15.4 69
5.1 Overview 69
5.2 An 802.15.4 LR-WPAN 71
5.3 802.15.4 Architecture 72
5.3.1 PHY Layer 72
5.3.2 MAC Layer 73
5.4 Time, Energy, and Bandwidth Related Limitations of 802.15.4 . . . 75
5.5 Enhancing 802.15.4 by Considering the Mentioned Limitations . . . 77
5.5.1 802.15.4 BED - A Proposed Scheme 78
5.6 Selection and Implications of Different Parameters 83
5.6.1 Relation between BO-BI and SO-SI 83
5.6.2 Maximum CFP Slots available 83
5.6.3 CFP Slots Actually Needed 85
5.6.4 Bandwidth Under-Utilization 87
5.7 Conclusions 87

6 AREA-MAC Protocol 89
6.1 Basic Concepts and Features 89
6.1.1 Limitations of the long preamble technique 89
6.1.2 AREA-MAC Overview 90
6.1.3 Transmission and Reception with AREA-MAC 91
6.1.4 AREA-MAC Features 95
6.2 AREA-MAC Design 102
6.2.1 Design Phases 103
6.2.2 Assumptions 103
6.2.3 Network Model 106
6.2.4 Traffic Model 108
6.2.5 Dissemination Model 109
6.2.6 Energy Model 109
6.2.7 Delay Model 112
6.3 Analytical Comparison of AREA-MAC with the Long Preamble Technique 112
6.3.1 Energy Gain 112
6.3.2 Delay Gain 115

6.3.3 Energy and Delay Gain Presentation 116
6.4 Conclusion 120

7 Performance Evaluation of AREA-MAC: A Simulation Approach 123
7.1 Simulation Setup 123
7.1.1 Simulation Model 124
7.1.2 Simulation Configuration 126
7.2 Evaluation Results 127
7.2.1 Normal Traffic Scenario 128
7.2.2 Burst Traffic Scenario 137
7.2.3 High Traffic (HT) Scenario 143
7.2.4 Varying Routing Effects 150
7.2.5 Scalability Effect - A Medium Size Network 157
7.2.6 Scalability Effect - An Extended Network 163
7.3 Discussion and Conclusion 170

8 Optimizing Energy and Delay With AREA-MAC Protocol 173
8.1 Duty Cycle Effect 173
8.1.1 Energy Consumption 174
8.1.2 End-to-End Delay 176
8.1.3 Packet Delivery Ratio 176
8.2 Optimizing Energy and Delay 178
8.2.1 Formulating Optimization Problems 178
8.2.2 Drawing Relationship Between Parameters 181
8.2.3 An Optimal Solution 182
8.3 Conclusions 183

9 Performance Evaluation of AREA-MAC: A Testbed Approach 185
9.1 Evaluation Setup 185
9.1.1 Parameters Setting 186
9.1.2 Packets Formatting 188
9.1.3 Testbed Deployment for the AREA-MAC Implementation . 190
9.2 Evaluation Results 190
9.2.1 Energy Consumption 190
9.2.2 End-to-End Delay 193
9.2.3 Packet Delivery Ratio 194
9.2.4 Preambles Transmitted 196
9.2.5 Preamble Received 197
9.2.6 Overheard Packets 197
9.3 Simulation Evaluation vs. Testbed Evaluation 198
9.4 Conclusion 201

10 Conclusion and Future Work **203**

10.1 Conclusion 203

10.2 Future Directions 205

10.3 Concluding Remarks 206

References **207**

List of Figures

1.1 Common Sensor Platforms . . . 2
1.2 Health Care Example . . . 5
1.3 WSN Requirements, Proposed Solutions, and Expected Outcome . . . 8

2.1 Node Architecture . . . 10
2.2 Network Architecture . . . 12
2.3 WSN Applications . . . 13
2.4 The Communication Protocol Stack . . . 15
2.5 Duty Cycling in WSNs . . . 21
2.6 Topology Control Example . . . 23

3.1 Contention Window Based CSMA Algorithm . . . 29
3.2 IEEE 802.11 DCF Based CSMA-CA Scheme . . . 30
3.3 Channel Accessing in WSNs . . . 31
3.4 Periodic Listen-Sleep Schedule with S-MAC . . . 32
3.5 Virtual Cluster Formation with S-MAC . . . 33
3.6 T-MAC . . . 35
3.7 The Working of STEM . . . 36
3.8 TRAMA . . . 39
3.9 Channel Polling in WSNs . . . 41
3.10 ALOHA with Preamble Sampling . . . 43
3.11 WiseMAC . . . 46
3.12 Basic Working of SCP-MAC . . . 49
3.13 Basic Working of Funneling-MAC . . . 50

4.1 OMNeT++ Module Hierarchy . . . 58
4.2 NIC Module of MF . . . 59
4.3 Architecture of the DES-Testbed . . . 61
4.4 MSB-A2 Hardware Details . . . 62
4.5 Box-and-Whisker Plot . . . 67

5.1 IEEE 802.15.4 Topology Structure . . . 71
5.2 IEEE 802.15.4 protocol architecture . . . 72
5.3 MAC Superframe Structure . . . 74

5.4 GTS Characteristics Field 75
5.5 GTS Descriptor 76
5.6 802.15.4 BED Scheme 79
5.7 Revised Beacon Frame 81
5.8 Revised GTS Characteristics Field 82
5.9 BI-BO and SD-SO Relation 84
5.10 Maximum Available CFP Slots 84
5.11 Data Packets and CFP Slots (with ACK) 86
5.12 Data Packets and CFP Slots (without ACK) 86
5.13 Bandwidth Under-Utilization Example 87

6.1 AREA-MAC Basic Working 91
6.2 Basic Format of an AREA-MAC Preamble 91
6.3 Radio Switching of AREA-MAC Nodes 92
6.4 State Transition at the Sending Node 94
6.5 State Transition at the Receiving and Forwarding Node 95
6.6 Transmission and Reception Flow Chart 96
6.7 Adaptive Duty Cycling 100
6.8 Different Roles of an AREA-AMC Node 106
6.9 WSN Topologies 107
6.10 Time and Energy Gain Comparison 117
6.11 Transmit Energy and Delay Gain of AREA-MAC 118
6.12 Receive Energy Gain of AREA-MAC 119
6.13 AREA-MAC Duty Cycle Gain 120

7.1 Overview of the Simulation Framework 124
7.2 Detailed Simulation Framework 125
7.3 Packets Format for Simulation 126
7.4 Snapshots of Grid and Random Topologies 128
7.5 Energy Consumption Comparison for the Simple Traffic Scenario . 130
7.6 End-to-End Delay Comparison for the Simple Traffic Scenario . . . 132
7.7 Packet Delivery Comparison for the Simple Traffic Scenario 133
7.8 Preamble Transmission/Reception for the Simple Traffic Scenario . 135
7.9 Overheard Packets Comparison for the Simple Traffic Scenario . . . 136
7.10 Energy Consumption Comparison for the Burst Traffic Scenario . . 138
7.11 End-to-End Delay Comparison for the Burst Traffic Scenario . . . 139
7.12 Packet Delivery Comparison for the Burst Traffic Scenario 141
7.13 Preamble Transmission/Reception for the Burst Traffic Scenario . . 142
7.14 Overheard Packets Comparison for the Burst Traffic Scenario . . . 143
7.15 Energy Consumption Comparison for the High Traffic Scenario . . 145
7.16 End-to-End Delay Comparison for the High Traffic Scenario 146
7.17 Packet Delivery Comparison for the High Traffic Scenario 147
7.18 Preamble Transmission/Reception for the High Traffic Scenario . . 149
7.19 Overheard Packets Comparison for the High Traffic Scenario 150

7.20 Energy Comparison Between AREA-MAC N_1 and N_2 Schemes . . 151
7.21 Delay Comparison Between AREA-MAC N_1 and N_2 Schemes . . . 152
7.22 Packet Delivery Ratio Comparison Between the N_1 and N_2 Routing Schemes 154
7.23 Preamble Transmission/Reception Comparison Between AREA-MAC N_1 and N_2 Schemes 155
7.24 Overheard Packets Comparison Between AREA-MAC N_1 and N_2 Schemes 156
7.25 Energy Comparison for the Medium Size Network 158
7.26 End-to-End Delay Comparison for the Medium Size Network . . . 159
7.27 Packet Delivery Comparison for the Medium Size Network 161
7.28 Preamble Transmission/Reception for the Medium Size Network . . 162
7.29 Overheard Packets Comparison for a Medium Size Network 163
7.30 Snapshots of Grid and Random Topologies of an Extended Network 164
7.31 Energy Consumption Comparison for an Extended Network 165
7.32 End-to-End Delay Comparison for an Extended Network 166
7.33 Packet Delivery Comparison for an Extended Network 167
7.34 Preamble Transmission/Reception Comparison for an Extended Network 169
7.35 Overheard Packets Comparison for an Extended Network 170

8.1 Energy Consumption Comparison for Different Duty Cycle Values . 175
8.2 End-to-End Delay Comparison for Different Duty Cycle Values . . 176
8.3 Packet Delivery Comparison for Different Duty Cycle Values 177
8.4 AREA-MAC Basic Working with Packet Durations 179

9.1 CC1100 WOR Timers vs. Current Consumption 186
9.2 CC1100 WOR Setting for the AREA-MAC Implementation 188
9.3 DES Testbed Packets Format Configuration 190
9.4 DES Testbed Deployment for AREA-MAC Implementation 191
9.5 Energy Comparison of MSB-A2 nodes under AREA-MAC protocol 192
9.6 End-to-End Packet Delay of MSB-A2 nodes under AREA-MAC . . 193
9.7 Packet Delivery Ratio of MSB-A2 nodes under AREA-MAC 195
9.8 Preamble Transmission with MSB-A2 nodes under AREA-MAC . . 196
9.9 Preamble Reception with MSB-A2 nodes under AREA-MAC . . . 197
9.10 Overheard Packets for MSB-A2 nodes under AREA-MAC 198

List of Tables

2.1 Hardware Specification of Different WSN Platforms 11

3.1 Comparison of Different WSN MAC Protocols 51

4.1 Hardware Details of the DES-Mesh Router 61
4.2 Hardware Details of the MSB-A2 Sensor Node 62

5.1 IEEE 802.15 WPAN Standards 70

6.1 Notations Used Throughout the Chapter 93
6.2 Parameters setting for the analytical evaluation 118

7.1 AREA-MAC Simulation Configuration 127

8.1 Energy and Delay - Optimal Solution 182

9.1 CC1100 Duty Cycle Approximation 187
9.2 Configuration of the AREA-MAC Testbed Implementation 189
9.3 AREA-MAC Simulation and Testbed Implementation Differences . 200

List of Abbreviations

ADC	Analog to Digital Converter
ASK	Amplitude Shift Keying
BER	Bit Error Rate
BI	Beacon Inteval
BO	macBeaconOrder
BPSK	Binary Phase Shift Keying
C4ISRT	Command, Control, Communications, Computing, Intelligence, Surveillance, Reconnaissance, and Targeting
CAP	Contention Access Period
CCA	Clear Channel Assessment
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CTS	Clear to Send
CW	Contention Window
DCF	Distributed Coordinated Function
DSSS	Direct Sequence Spread Spectrum
ED	Energy Detection
FDMA	Frequency Division Multiple Access
FFD	Fully Function Devices

FIFO	First In First Out
FSK	Frequency Shift Keying
GFSK	Gaussian FSK
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GTS	Guaranteed Time Slot
HR-WPAN	High Rate WPAN
HRT	Hard Real-Time
I/O Interface	Input Output Interface
IFS	Inter Frame Space
IQR	Inter-Quartile Range
ISM band	Industrial, Scientific, and Medical band
LIFS	Long IFS
LLC	Logical Link Control
LQI	Link Quality Indication
LR-WPAN	Low Rate WPAN
MAC	Medium Access Control
MCPS-SAP	MAC Common Part Sublayer SAP
MCU	Micro-Controller Unit
MEMS	Micro Electro-Mechanical Systems
MHR	MAC Header
MLME	MAC subLayer Management Entity
MOUT	Military Operations in Urban Terrain
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
MSK	Minimum Shift Keying
NAV	Network Allocation Vector

NFS	Network File System
O-QPSK	Offset Quadrature Phase-Shift Keying
OOK	On-Off Keying
OSI	Open Systems Interconnection
PAN	Personal Area Network
PD-SAP	PHY Data SAP
PHR	PHY Header
PLME-SAP	Physical Layer Management Entity SAP
POS	Personal Operating Space
PPDU	PHY Protocol Data Unit
PQI	Preamble Quality Indicator
PSDU	PHY Service Data Unit
PSSS	Parallel Sequence Spread Spectrum
QoS	Quality of Service
RFD	Reduced Function Devices
RNG	Random Number Generator
RSSI	Received Signal Strength Indication
RTS	Request to Send
SAP	Service Access Point
SD	Superframe Duration
SHM	Structural Health Monitoring
SIFS	Short IFS
SNIR	Signal to Noise and Interference Ratio
SNR	Signal to Noise Ratio
SO	macSuperframeOrder
SPDU	SSCS Protocol Data Unit
SRD	Short Range Device

SRT	Soft Real-Time
SSCS	Service-Specific Convergence Sublayer
SWOR	WOR Strobe
TDMA	Time Division Multiple Access
WBAN	Wireless Body Area Network
WMN	Wireless Mesh Network
WOR	Wake-On-Radio
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

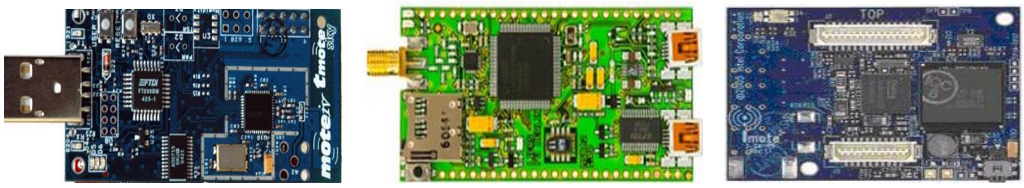
Nothing is predestined: The obstacles of your past can become the gateways that lead to new beginnings.

–Ralph Blum

THE evolution of computing and communication technologies has been exceptional during the last decades. Starting right from the vacuum tube technology in the first generation of computing in 1940s to the invention of transistors in 1950s and further to the introduction of micro-processors and large scale integration in 1970s, the consistent compliance of Moore's and Bell's Laws has always been sensed. Intel co-founder Gordon Moore in 1965 predicted that the number of transistors incorporated in a chip will approximately double every two years. Based on this premise, Gordon Bell in 1972 envisioned that after every decade we would have a new class of computing. This constant advancement, especially in the integration scale during the current fifth generation, has earned almost everything for the computing and communication world; size is shrinking, cost is reducing, switching power consumption is going down, speed and efficiency are going up, and the mobility and portability are easing up many barriers. Based on the pace at which the silicon technology is swiftly evolving, one may foresee a fundamental shift in this domain even before ten years.

The introduction of wireless networks, and then the integration of sensors has further revolutionized this domain by making distance, movement, and monitoring seamless. Wireless networks offer a great amount of flexibility by allowing users to get connected to the networks anytime and anywhere and unleashing from the restriction of costly and messy cables. *Embedded systems* can now be integrated into the environment, while keeping themselves invisible, to assist users in performing tasks that otherwise might be left undone. The smart processors can easily be found in almost any device around us, whether it be a home appliance or an office tool. This advancement, accompanied with the efforts to make small devices organize themselves, has introduced the domain of Wireless Sensor Networks (WSNs) [1–8].

The pervasiveness and self-organization of low-cost, low-power, long-lived, and small-sized sensor nodes[1] have brought a new perspective to the world of wireless communication and computation. This emerging paradigm is destined to play a huge role to our future ubiquitous world as it extends the reach of cyberspace out into physical and biological systems. Coupled with sensing, computation, and communication into a single tiny device, WSNs are emerging as an ideal candidate for several daily-life applications, particularly in monitoring and controlling domains. Demands placed on these networks are expending exponentially with the increase in their dimensions. The development of new technologies and continuous refinements of current approaches is also pushing this domain even further. Besides the development of new algorithms and protocols, many commercial hardware vendors and research institutes are also engaged designing novel and efficient architectures for sensor nodes. Figure 1.1 shows some of the sensor nodes used for deployment, experiment, and evaluation of different WSN related applications.



(a) Tmote Sky [9] (b) MSB-A2 [10] (c) Imote-2 [11]

Figure 1.1: Some of the common sensor platforms, which are being used by industrial and research organizations for several WSN related applications and testbeds. They differ from each other in processing, storage, and communication capabilities and are suitable for an application or the other.

The underlying theme of this thesis is to design a MAC protocol for WSNs. Since WSNs can pose very complex design challenges due to very limited resources available to sensor nodes, especially energy, the proposed MAC protocol aims to provide energy efficient operations for WSNs. Additionally, it reduces network latency and provides high packet delivery for an asynchronous and adaptive WSN. This chapter gives an overview of the motivation, contribution, and thesis organization.

1.1 Background and Motivation

As communications and computing devices continue to proliferate, innovative medium access techniques are being developed to efficiently share the limited wireless bandwidth. Like other wireless networks, WSNs share a wireless broadcast medium

[1]The terms sensor node, node, wireless node, smart node, and mote are used somehow interchangeably.

among sensor nodes. However, the unique requirements and challenges of WSNs make them quite different from other wireless networks, and therefore most of their advances cannot be directly applied *out of the box* to WSNs [12]. Very limited resources, often dense deployment, collaboration among sensor nodes, dispersed applications, and volatile communication links of WSNs need unique medium access techniques. These demands, however, have made the MAC design for WSNs challenging yet an interesting field for the researchers, and as result to that, a high proliferation of channel accessing techniques and protocols for WSNs over the last years has been witnessed [13–17].

The most substantial challenge facing these networks is the requirement of significant reductions in energy consumption of sensor nodes. Size and cost limitations result in a very limited on-board power capacity for sensor nodes. For example, the SunSPOT platform [18] has 770 mAh lithium-ion battery capacity and the Smart Dust mote [19] has 33 mAh capacity. The MSB-A2 platform [10] and the Imote2 [11], MicaZ [20], Mica2 [21], IRIS [22], and TelosB [23] platforms from the Crossbow (now Memsic [24]) can use two AA batteries, which provide a typical capacity of 2500 mAh and can supply up to the conservative estimate of 2200 mAh at 3 Volts [25]. If sensor nodes are running in 'always on' mode, they could drain this energy within 3-4 days. The remote and inaccessible deployment of sensor nodes prevents recharging their batteries, making energy a scarce resource for WSNs. A WSN MAC protocol can substantially conserve the energy by controlling the functionalities of the radio, which is the major energy consumer component of a sensor node [1, 19, 26]. Consequently, to increase the network lifetime of a WSN, the MAC protocol usually trades performance and fidelity in favor of energy.

Different MAC protocols proposed for WSNs use different working space but are mainly tailored towards the *single* prime objective of energy efficiency. Other performance criterion such as latency, scalability, asynchrony, packet delivery, and/or adaptability to traffic and topology changes are mostly overlooked or dealt as secondary objectives. Some of the proposed MAC protocols consider that all nodes in a WSN are time-synchronized and wake up and sleep simultaneously [27–29], whereas other protocols assign time/frequency slots to each sensor node and force the synchronized nodes to access only their allocated slots [30–32].

With the advent of new MEMS and energy harvesting/scavenging technologies, and with the dynamic increase in several WSN related applications, the research community is set to experience many diverse directions in this domain. Though the amount of energy extracted from the current energy harvesting technologies is still limited, having witnessed the significant technology advancements, particularly in the hardware domain, one may expect better results in the near future. Therefore, we believe that to efficiently utilize WSNs on a large scale for a long term and in a wide range of applications, energy efficiency, even though being the most critical metric, is not sufficient to address. Other performance criterion, especially latency, can play a critical role in many WSN applications. Appraising and leveraging the energy implications are important from the future perspective. Synchronization in low-resource sensor nodes results in high energy consumption, overhead, and complexity. Forcing nodes to use only the allocated slots, on the one hand may reduce energy consumption for sensor nodes, but on the other hand, it results in higher complexity and queuing delays, since each node has to wait for its turn to access the channel. Moreover, both the synchronization and scheduling restrict the network scaling and adaptability up to some limit.

Some of the mentioned metrics may temporarily outweigh energy efficiency as per requirements of the application. For medical, surveillance, security, terrorist attacks, home automation, flood, fire, and seismic detection applications, the provision of timeliness is as crucial as saving energy. Researchers foresee a vital role of WSNs in health related applications [33–35]. In clinical diagnostics, traditional paper based monitoring systems for patients are complex, expensive, and time consuming. The increase in the world population, diseases, and unfortunate incidents, and the inadequate number of doctors and clinicians available around highlight the need of an automatic system. WSNs can fulfill such requirements, since tiny and cheap sensor nodes can be seamlessly integrated into WPANs or WBANs.

An example of such applications is shown in Figure 1.2, where a patient is wearing an e-textile integrated with sensor nodes. Nodes can be used to measure different biological parameters and to remind medical staff for their scheduled check-ups regardless the physical location of the patient or medical staff. In such applications, energy is a factor to be efficiently handled in order to increase the system lifetime. However, the grandness of timeliness and packet reception also increases sharply. Sensor nodes should automatically but timely alert doctors or emergency services when the patient suffers from a severe disease or any other mishap happens. The importance of delivery ratio may also increase as one disease could immediately result in other diseases, and the sensor nodes deployed on or near different body parts need to urgently inform medical personnel about the relative body part.

Such systems can also be used to store patient records for future research and automatic medication purposes, or to monitor the behavior of small animals both on land and in sea. The idea behind the large efforts of the automotive industry to have *interconnected cars* also needs a timely information available to drivers about the road accidents and other traffic conditions. Moreover, energy and delay efficient sensor nodes are the real driving force behind the emerging concept and construction of *Internet of Things (IoT)* [36–38] that aims to interconnect trillions of everyday objects to the Internet. The European Union has recently officially endorsed the development of the IoT.

1.2 Thesis Contribution and Organization

This thesis is mainly dedicated to the design of the Asynchronous Real-time Energy efficient and Adaptive MAC (AREA-MAC) protocol for WSNs. AREA-MAC efficiently reduces energy consumption through dealing with collision, idle listen-

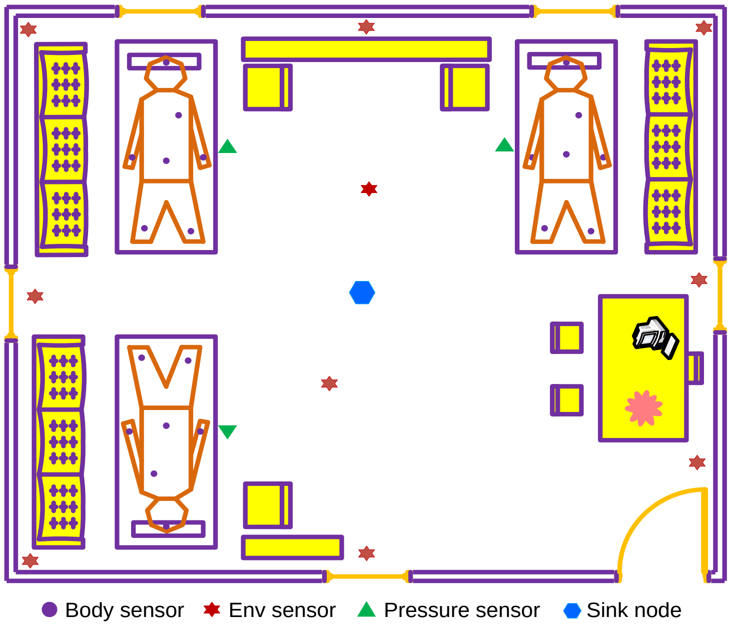


Figure 1.2: An example of a health-care scenario. Body sensor nodes are attached to the patient body to measure different biological parameters such as heart and pulse rate, blood pressure, ECG, and/or EEG values. Environment nodes measure different physical parameters such as temperature, air pressure, humidity, and light values. Pressure nodes are deployed under or near the patient bed to timely alert medical staff for an unexpected downfall of the patient. The sink node is responsible to collect and forward the traffic generated by sensor nodes to the end-user.

ing, over-hearing, and over-emitting. It also reduces packet latency by enabling the receiving node to respond early and adaptively to the sending node. Furthermore, unlike many other MAC protocols dedicated to operate over WSNs, AREA-MAC does not need any sort of synchronization, time/frequency scheduling, and clustering among sensor nodes, which greatly reduces overhead and energy consumption for sensor nodes and helps a WSN to scale and adapt to network conditions rather smoothly. The problem statement of this research work mainly centers around the following issues:

- When and for how long radios of sensor nodes need to wake up, so that energy consumption and end-to-end packet delay of a WSN can be reduced and the packet delivery ratio can be kept up to an acceptable level?
- What is the trade-off between energy and delay factors? What is their fall-out on other performance criterion?
- How do the changes in traffic conditions, topology, and scalability affect the performance of an asynchronous WSNs?

There are nine chapters that follow this introduction. They are organized as follows. **Chapter 2** gives a brief yet comprehensive insight into the WSN domain, where we first discuss the architecture, applications, design objectives, and challenges of WSNs. Afterwards, an overall understanding of the design aspects of MAC protocols for WSNs has been presented, where we discuss why traditional MAC protocols are not well suited to WSNs by outlining a set of important MAC attributes. We show that in order to provide energy efficient operations, a MAC protocol for WSNs must expeditiously deal with idle listening, overhearing, collisions, over-emitting, and control packet overheads and to support low packet latency, it has to adapt well to the network conditions.

Afterwards, in **Chapter 3**, we elaborate state of the art and classify the existing MAC protocols proposed for WSNs into the four major categories of *contention based*, *scheduling based*, *channel polling based*, and *hybrid protocols* [39]. We present the detailed working, strong, and weak points of several well-known MAC protocols falling in each category. Although the channel polling scheme consumes less energy than other schemes, it still suffers from several limitations resulting from the use of the extended preamble. For our proposed scheme, we conciliate these limitations by using short preambles and several other methods.

At the end of the chapter, we compare all discussed protocols for their support for energy, timeliness, asynchrony, adaptivity, and scalability parameters in a tabular form. We conclude that most of the proposed MAC protocols are suboptimal for a given parameter. They either partially support a parameter or trade-off with another parameter, especially in favor of energy. Importantly, most of them do not consider latency as an important metric in their design, which motivates the need of an energy as well as delay efficient MAC protocol for WSNs.

Later on, in **Chapter 4**, we discuss the evaluation methods and metrics that have been used to analyze and evaluate our contributions. The capability or worth of any system/protocol is validated usually through three types of evaluation methods, namely *analytical modeling*, *simulation*, and *real experimentation* methods [40, 41]. In our work, we use all three types of evaluation methods, and in addition we also use the optimization method to find the optimal duty cycle value for sensor nodes that could facilitate minimum energy consumption and packet latency. The brief introduction of the simulation and testbed platforms has also been discussed in this chapter. The metrics of energy consumption, end-to-end delay, packet delivery ratio, preamble transmission, and overheard packets that we use to evaluate our proposed scheme have also been briefed. At the end of the chapter, we elucidate the way result graphs obtained from the simulation and real experimentation platforms are drawn.

Chapter 5 and onwards start our contribution. In **Chapter 5**, we critically analyze the emerging IEEE 802.15.4 protocol [42] and rectify its several limitations, particularly for energy, delay, and bandwidth related issues. We propose a simple scheme called *802.15.4 BED* to overcome those limitations and outline an application-specific parameters setting for energy, time, and bandwidth critical WSN applications by using the analytical approach. The early steps of implementing an 802.15.4 prototype have been started at Technische Universität Berlin, by using Motiev Tmote nodes [43] under Tinyos2.x [44] platform. After switching over to Freie Universität Berlin and with the consensus of supervisor, we reconsidered our research objectives and started focusing on the design, implementation, and evaluation of new MAC (AREA-MAC) protocol for WSNs. Therefore in this chapter, we have only presented the proposed methodology and the related analytical results of 802.15.4 protocol and leave experimentation results as a future work.

Subsequently, in **Chapter 6**, we revisit the energy, timeliness, adaptability, synchronization, and redundancy problems of WSNs by proposing AREA-MAC protocol. AREA-MAC uses the channel polling scheme [39] to access the channel. However, unlike the traditional channel polling scheme [45–47], which uses a very long preamble preceding to each data packet, AREA-MAC uses the short preamble technique. The sender sends out a stream of short preambles with a short spacing and turns its radio to receive mode in between in order to receive the *pre-ACK* from the intended receiver. The sender sends the data packet as soon as it receives the pre-ACK. This minimizes the chances of the data packet being dropped as the sender and receiver communicate via a 'pre-established' link. Each short preamble contains information such as the source address, destination address, sequence number, and message type of the forthcoming data packet and the *data-to-follow* bit to inform the receiving node whether the sender has more data packet(s) waiting in its queue to be transmitted.

Short preambles unified with the pre-ACK mechanism (by using the source and destination addresses) impede several problems of the long preamble technique such as unnecessary energy consumption both at the receiver and sender ends, excessive latency at each hop, increased collision probabilities with the increased transmission and reception lengths, overhearing at non-targeted receivers, and bandwidth waste on the broadcast medium. The sequence number of the forthcoming data packet present in the preamble helps sensor nodes in reducing the number of redundant data packets. AREA-MAC considers different message types to support both normal and real-time traffic, which is indicated by the message type information of the preamble. After transmitting a preamble and during the waiting time to receive the pre-ACK, if the node receives a real-time packet, it drops its bid and starts immediately responding the sending node. The receiving node can also adapt its wake-up schedule in accordance with the sender, if the sender has more data packets in its queue indicated by the *data-to-follow* bit, which further decreases packet latency and energy consumption of sensor nodes.

All these features of AREA-MAC, beside several other, detailed in Chapter 6, greatly improve energy consumption, packet latency, and packet delivery for a WSN. Figure 1.3 demonstrates the main features of AREA-MAC by aligning them with the overall design objectives and challenges of WSNs (discussed in Section 2.3) and with the expected outcome. We also mathematically drive network, traffic, energy, and delay models for AREA-MAC in this chapter. The analytical comparison of AREA-MAC with the long preamble technique has been presented, which shows significant improvement in terms of energy and delay for AREA-MAC nodes.

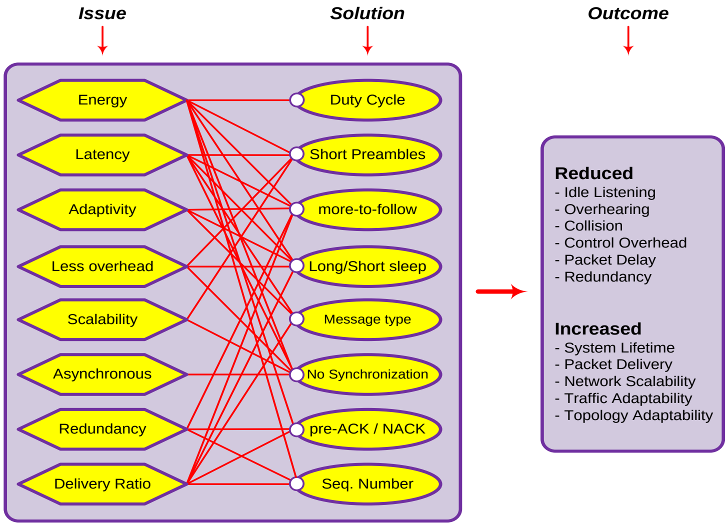


Figure 1.3: The overall design objectives/issues of WSNs, our contribution, and the improved outcome.

Later on in **Chapter 7**, the simulation implementation of AREA-MAC based on the OMNeT++ simulation platform [48] has been carried out and discussed. After discussing the simulation model used for the evaluation, we present several results related to energy, delay, packet delivery, preamble transmission, and packet overhearing, which show significant improvement of AREA-MAC over the long preamble scheme. Additionally, the effects of varying topology and traffic conditions, broadcast vs. unicast communication, scalability, cross layering, and different duty cycle values are also analyzed.

In the subsequent **Chapter 8**, we formalize three different optimization problems for AREA-MAC. These optimization problems correspond to minimization of energy consumption, minimization of end-to-end delay, and maximization of overall system lifetime. The optimization problems related to energy and delay have then been solved for the different duty cycle values, and their optimal solutions have been discussed and matched with the simulation results.

After analytical and simulation evaluations, we, in **Chapter 9**, present the implementation and evaluation of AREA-MAC on the DES-Testbed [49]. The DES-Testbed is a hybrid wireless testbed deployed across the different building campuses of Freie Universität Berlin. After briefly outlining the topology structure of the DES-Testbed and implementation methods, we discuss several implementation results. We compare the testbed results with that of the simulation results and find out the differences between both platforms. We try to ascertain the reasons for their differences and discuss ways to narrow down their gap.

Finally, **Chapter 10** concludes the research work presented in this thesis and figures out the main contribution by highlighting the advantages of the proposed scheme. It also outlines directions for the future work.

Chapter 2

Wireless Sensor Networks

The only way of finding the limits of the possible is by going beyond them into the impossible.

–Arthur C. Clarke

THE emerging domain of WSNs has lately received a tremendous attention from both academia and industry, which has transferred it from obscurity into the reality. A considerable amount of research activities focuses on the issues that explore new directions for WSNs at the component, system, and application levels. This domain is destined to revolutionize the way we live, behave, and interact with the physical world by detecting, capturing, and converting events into a form that can be processed and rightly acted upon. This chapter first enlightens some of the salient features of WSNs, and then the important design aspects of their MAC protocols are outlined.

2.1 Architecture

A WSN usually consists of several sensor nodes distributed either inside or very close to a geographical region of interest with a view to sense, collect, and disseminate data relating to one or more parameters. The architecture of the WSN can reasonably be divided into the *node architecture* and *network architecture*. Energy efficiency can be achieved at both node and network levels [26]. At the node level, radio management, modulation, computation, packet forwarding, and interaction among layers can be made energy efficient. At the network level, energy aware topology and traffic management, better collaboration and communication among sensor nodes, and the reduced overhead can greatly help in the energy efficient objective of the WSN. Both of these architectures are detailed below.

2.1.1 Node Architecture

A sensor node mainly consists of five basic components; an *MCU unit*, a *radio unit*, a *memory unit*, an *I/O interface unit*, and a *power unit*. Figure 2.1 depicts the basic architecture of the sensor node, whereas Table 2.1 captures more specific hardware details in terms of these components for the sensor platforms shown in Figure 1.1. The low-power MCU usually consists of a microcontroller or a microprocessor, which provides intelligence to the node by performing tasks, processing data, and controlling the functionality of other components. The sensor node usually comes with the self-sufficient and cost-effective microcontroller with the integrated memory unit. For better power management purposes, the MCU may support different operating modes such as an active, idle, and sleep mode. Widely used MCUs are the StrongARM microprocessor family (currently replaced with XScale series) from Intel [50], LPC mircocontroller series from NXP Semiconductors [51], MSP series from Texas Instruments [52], and AVR microcontroller from Atmel [53].

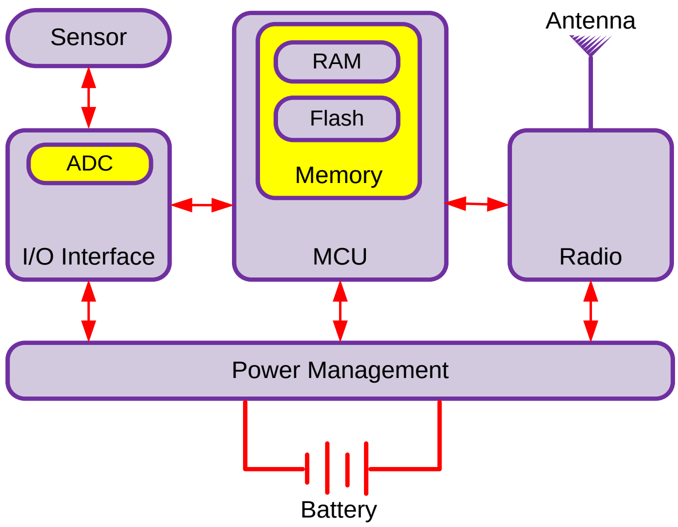


Figure 2.1: Architecture of a typical sensor node. It usually includes a power management system, which is very important component to manage the consumption of power. Sensors attached to the I/O interface are usually used to measure temperature, humidity, air pressure, or light values of a physical object or phenomenon.

The radio transceiver contains an antenna, frequency synthesizer, oscillator, demodulator, amplifier, and other circuitry needed to communicate with other sensor nodes over the radio channel. Like the MCU, the radio can also operate in different operating modes such as transmit, receive, idle, and sleep mode. The radio is an important component, especially for the energy efficient operations of the sensor node. It helps in deciding several factors such as power consumption, carrier frequency, data rate, modulation, coding schemes, transmission power, SNR and RSSI values, error blocking, and many more [3]. Commonly used radio chips

Table 2.1: Detailed hardware specifications of the WSN platforms shown in Figure 1.1.

	Tmote Sky	**MSB-A2**	**Imote-2**
CPU	TI MSP430	NXP LPC2387	PXA271 XScale
• Speed	8 MHz	up to 72 MHz	13 - 416 MHz
Radio	CC2420	CC1100	CC2420
• Frequency	2.4 GHz	315/433/868/915 MHz	2.4 GHz
• Data Rate	250 kbps	up to 500 kbps	250 kbps
• RX Current	18.8 mA	15.6 mA	18.8 mA
• TX Current	17.4 mA	28.8 mA	17.4 mA
• Modulation	DSSS	2-FSK/GFSK/MSK/ OOK/ASK	DSSS
Memory			
• RAM	10 KB	98 KB	32 MB
• Flash	48 KB	512 KB	32 MB

are the several versions from Chipcon (now Texas Instrument) [54], MPR2400 used by Mica series [55], and the RFM TR Series [56] radios.

The power unit supplies battery power to drive all other components of the sensor node. Due to its limited capacity, energy aware operations by each component are required. The I/O interface unit integrates several application-specific sensors, which observe a physical phenomenon and generate traffic based on the observed phenomenon. This component may include an ADC to convert analog signals produced by sensors into digital one before feeding them to the processing unit.

In addition to these basic components and as per requirements of the application, the sensor node can also be equipped with some additional components. For some localization applications, needing the knowledge of the physical location, the node can have a GPS component attached to it. For mobile and mechanical related applications, a motor or a mobilizer may be attached to move sensor nodes. To have an increased power supply, nodes can carry an additional power generator with them, which may utilize solar, thermal, kinetic, or vibration energy to generate extra power. However, such components usually require too much power or are too heavy to be practicable for low-power and light-weight matchbox-sized sensor nodes.

2.1.2 Network Architecture

Several scattered nodes in a sensor field communicate and collaborate with each other in an ad hoc fashion to form a WSN. Each of these sensor nodes has dual responsibilities of generating and routing data back to the sink node, usually via multi-hop paths. Figure 2.2 shows a typical network architecture of a WSN. The sink node may communicate with the end-user via a gateway by using the Internet or any other communication network, so that the disseminated data can be stored, treated, and analyzed. The sensor field can have more than one sink nodes depending on the terrain, size, and traffic load of the field. The gateway can also be connected to more than one WSNs, where sensor nodes may perform totally different tasks.

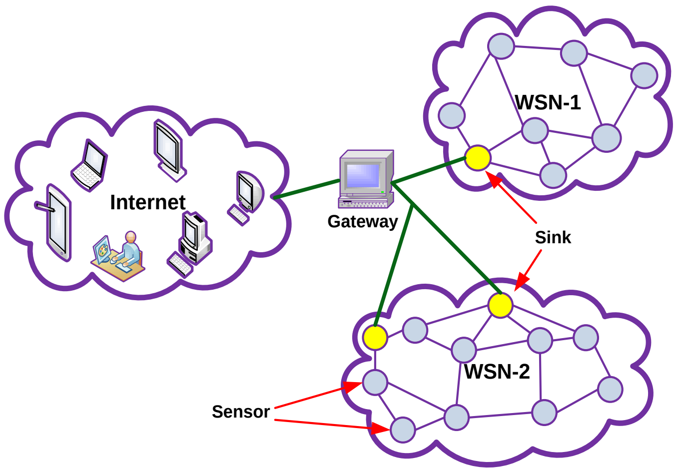


Figure 2.2: Architecture of a typical WSN. The capabilities of sensor nodes in the WSN may vary widely. Some nodes may perform simple tasks of monitoring a single phenomenon, whereas the other may perform complex and multiple sensing or aggregating tasks.

The network architecture of WSNs, depending on how sensor nodes communicate with each other, can be further divided into *flat architecture* and *hierarchical architecture* [57]. In flat architecture, each sensor node is a peer and has the same capabilities in performing a sensing task. Sensor nodes form multi-path routes to the sink node in a distributed fashion by relaying data to other peers. In hierarchical networks, nodes are organized into clusters, each one is supervised by a cluster head. The cluster members send their data to the head, which then relays it to the sink node in a single- or multi-hop manner. The cluster head may have different capabilities than other nodes. Both of these sub-architectures have their own advantages as well as disadvantages.

2.2 Applications

The application domain of WSNs is very diverse and broad, which includes, but not limited to, military, environmental, health, home automation, industrial, smart space, and security related areas [1, 7, 2]. Figure 2.3 wraps up several application domains of WSNs. WSNs can help to avoid catastrophic infrastructure disasters, facilitate C4ISRT military systems, conserve precious natural resources, increase productivity, monitor bridges, enable new smart home technologies, and in many other countless applications. To name a few, WSNs have been used to:

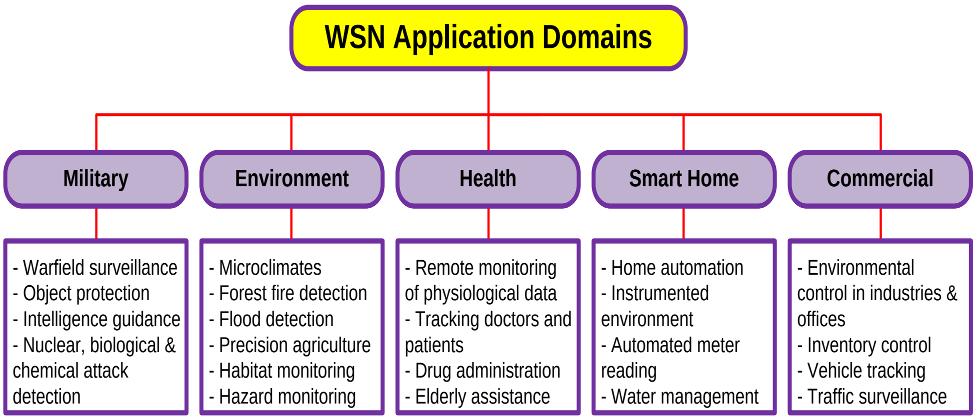


Figure 2.3: A wide range of applications shows the diversity of the WSN domain. This diversity keeps increasing as the technology advances over time.

- create a world wide IoT network to form a new information ecosystem for objects and people under the on-going Central Nervous System for the Earth (CeNSE) project from HP Labs [37] and the Smarter Planet Campaign from IBM [38]
- monitor air temperature, relative humidity, and photo-synthetically active solar radiation in a 70-meter tall redwood tree [58]
- gather the findings of latest wildfire monitoring system to analyze the social co-location patters of European badgers (Meles meles) [59]
- measure temperature and humidity in potato plants [60]
- observe nesting burrows of storm-petrel sea birds at Great Duck Island [61]
- detect intrusion at MacDill Air Force Base in Tampa, Florida [62]
- detect acoustic signals of a gun shot, and locate shooters at the MOUT facility of US Army [63]
- assist military and surveillance related missions under the Smart Dust [64] and VigilNet [65] projects
- investigate interaction and movement of Zebras in Kenya [66]
- collect seismic information based on earthquakes near the active volcano, Volcan Reventador, in Ecuador [67]
- detect early flood in Honduras in order to minimize its effects on urban life [68]
- monitor patients via wearable sensors under the CodeBlue project of the Harvard University [69]

- estimate the water flow with Non-intrusive Autonomous Water Monitoring Systems (NAWMS) [70]
- track the spatio-temporal vibration patterns in SHM applications [71]

2.3 Design Objectives and Challenges

While WSNs share the wireless medium among sensor nodes like other wireless networks, they are subject to a variety of unique objectives, challenges, and constraints, which considerably distinguish them from their counterparts [12]. Some of the main design objectives of sensor networks are concisely outlined below [57]. Note that the application-specific nature of the WSN does not need to implement all of the objectives at one instance.

- *Small node size* to allow dense deployment in harsh and hostile environments
- *Low node cost* to reduce overall cost of a dense network
- *Low power consumption* to prolong network lifetime
- *Application diversity* to make WSNs suitable for several applications
- *Self-configurability* among sensor nodes to autonomously organize themselves, even under varying traffic and topology situations and un-engineered deployment
- *Scalability* to support different network sizes under different applications and conditions
- *QoS oriented* to behave in terms of delay and reliability according the requirements of the application
- *Simplicity* to run uncomplicated yet efficient algorithms
- *Adaptability* to face varying traffic and post-deployed topology situations
- *Reliability* to deliver data efficiently under harsh and varying topological conditions
- *Fault Tolerance* to enable sensor nodes for the automatic repair and recovery process in an unattended environment, if the faint hardware gets failed or blocked for a while
- *Convergecast ability* to support flowing of data in many-to-one pattern, i.e., from nodes towards the sink node

All these objectives, along with the limited resources available to sensor nodes, pose several challenges in the design of sensor networks. Integrating sensing, processing, and communication functionalities into a sensor node has added a lot of complexities. Moving from sensors with only few hours of lifetime to one with many years of life time demands several iterations of energy efficient techniques. Shrinking size of nodes requires small size transceivers. Mapping the overall system requirements down to individual device capabilities is not an easy task. The massive and random deployment obligates sensor nodes to self-organize themselves over an inaccessible, dynamic, and unreliable environment. The direct interaction of WSNs with the real world and the application-specific nature require them to respond accordingly. As a result, a detailed understanding of capabilities, requirements, constraints, and limitations of WSNs is required.

Having gone through the comprehensive insight into the several important aspects of wireless sensor networks, we next discuss the importance, challenges, and common approaches related to the MAC designing for such networks. Most of the work presented below (and also in the next chapter) has been taken from our book chapter [13].

2.4 MAC Basics for WSNs

The MAC sublayer is a part of the data link layer specified in the communication protocol stack and is shown in Figure 2.4. It provides the channel access mechanism to several medium sharing devices. On a wireless medium, which is shared

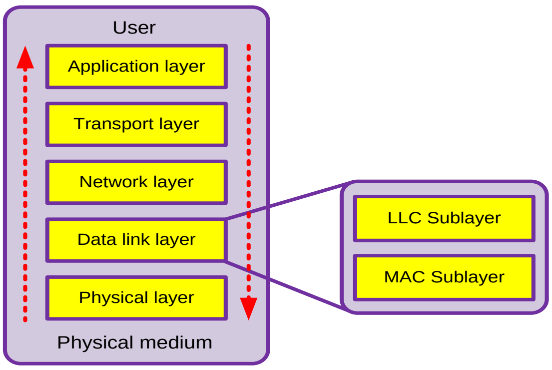


Figure 2.4: The communication protocol stack. This five-layered simplified model is commonly applied to network research as apposite to the seven-layered OSI model. An end-user can use application specific software/algorithms at the application layer. The transport layer helps maintaining the sensor data flow. The network layer routes data on an appropriate path. The LLC sublayer of the data link layer provides framing, flow control, error control, and link management facilities, whereas the MAC sublayer manages collisions and helps in energy aware operations of sensor nodes. The physical layer takes care of the radio, channel, modulation, transmission, and reception of bits on a physical medium.

by multiple devices and is broadcast in nature, when one device transmits, every other device in the transmission range receives its transmission. This could lead to an interference and collision of the frames when a transmission from two or more devices arrives at one point simultaneously. Sensor nodes usually communicate via multi-hop paths over the wireless medium in a scattered, dense, and rough sensor field. A MAC protocol manages the the communication traffic on a shared medium and creates a basic network infrastructure for sensor nodes to communicate with each other. Thus it provides a self-organizing capability to nodes and tries to enforce the singularity in the network by letting the sender and receiver communicate with each other in a collision- and error-free fashion.

Moreover, the typical requirement to increase lifetime of a WSN without the need of any power replacement and/or human interaction has prompted the development of novel protocols in all layers of the communication stack. However, prime gains can be achieved at the data link layer, where the MAC protocol directly controls the activities of the radio, which is the most power consuming component of resource-scarce sensor nodes. Efficient MAC protocols utilize the radio judiciously to conserve its energy. Thus the MAC protocol helps fulfilling important design objectives of WSNs by specifying how nodes employ the radio, share the channel, avoid collision in correlated and broadcasting environments, response the inquirer timely, and survive for a longer period. Hence, designing novel solutions for MAC protocols for WSNs has been and will remain a focal point for many researchers.

2.4.1 MAC Services

In general, the fundamental task of any MAC protocol is to regulate the *fair access* of sensor nodes to the shared medium in order to achieve good individual throughput and better channel utilization [72]. However, constrained resources, redundant deployment, and collaboration rather than competition among nodes considerably change the responsibilities of the MAC protocol for WSNs. On one hand, some relaxations may be granted to such MAC protocol. For example, nodes in WSNs usually send very small frames and use the channel occasionally, either periodically or whenever an important event occurs. Fairness in WSNs is not as important as in other networks, since nodes cooperate to achieve a common purpose. They remain idle or in sleep mode most of the time and rarely compete for the channel. Achieving good channel utilization is usually not considered as an important metric for WSNs. The data flow in WSNs is usually unidirectional, i.e., from nodes to the sink node, and end-users generally focus on the collective information rather than the individual throughput.

On the other hand, the MAC protocol for WSNs has some extra responsibilities to deal with as well. First and foremost is the issue of energy conservation. Since a distributed network of several nodes demands for long-time and maintenance-free operations, a MAC protocol -irrespective to the scheme and work space it uses- certainly must have built-in power-saving mechanism. Along with energy efficiency and as per application requirements, provision of timeliness, adaptability to traffic and topology conditions, scalability, support for non-synchronized operations, and interaction with other layers via cross-layering may also play an important role in designing the MAC protocol for WSNs. Furthermore, due to the dense deployment and small transmission distances, the transmission power of sensor nodes can often be lesser than the receiving power. It is unlike other traditional networks, which usually consider the receiving power to be negligible. Therefore, the WSN MAC protocol has to preferably consider a balanced equation between transmitting and receiving packets. Additionally, the ideal MAC protocol ensures self-stabilization, graceful adaptation, an acceptable delivery ratio, low overhead, and low error rates for a WSN.

2.4.2 MAC Challenges

The design of the MAC protocol for WSNs is a complex task due to the energy constraints, low transmission ranges, and compact hardware design of sensor nodes. Along with these factors, the event- or task-based network behavior and application diversity of WSNs also demand for peculiar MAC schemes, which are not common with traditional wireless networks. Additionally, by virtue of the wireless broadcast medium, WSNs inherit all the well-known problems of wireless communication and radio propagation in the shape of interference, fading, path loss, attenuations, noise, and high error-rates [73, 74]. The MAC protocol is directly influenced by all these effects as it sits right above the physical (PHY) layer, thereby presuming total control over the medium and its access rate. The frequent use of the unlicensed ISM band for most sensor networks and sensing applications can also worsen these effects.

The need of energy efficient operations for WSNs usually results in trades-off among energy and other parameters such as latency, packet reception, throughput, fairness, and scalability. As a result, many existing architectures and protocols for traditional wireless networks such as IEEE 802.11 and Bluetooth are not suitable for WSNs, as they usually target higher data rates with less emphasis on energy and other WSN specific issues. Among the three basic responsibilities of WSNs, namely *sensing*, *processing*, and *communication*, each performed by the sensors, MCU, and radio respectively, the later is the major power consumer [1, 19, 26]. The sensing power, though depends on the type of sensors used, working of ADC, nature of the application, and the complexity of event detection, is significantly less than the communication power. Many *passive sensors* such as temperature, light, humidity, and seismic sensors consume almost negligible power as compared to the *active sensors* such as sonar rangers, mobilizer, and the camera sensors with pan-zoom-tilt requirements [26].

Similarly, the power needed to process data is also much less than the communication power. Authors in [19] conclude that the transmission of a 1 kb packet over a distance of 100m is approximately equal to executing 3 million instructions by a typical microprocessor, which gives a hint on the utmost difference between the two powers. The ratio of communication to computation power has been calculated between 1500 and 2700 for Rockwell WIN sensor nodes in [26] and between 1000 to 10000 (for one bit of data) in [75].

The communication power of the node depends on several factors that include the type of modulation scheme used, data rate, transmit power, operational modes of the radio, and the switching frequency between these modes. At the same time, a MAC protocol can be made accountable for the following sources of energy waste in WSNs, which mainly relate to the communication [1, 15].

- **Idle listening:** Since a node in a WSN usually does not know when it will be the receiver of a message, it keeps its radio in ready-to-receive mode, which consumes almost as much energy as in receive mode. In low traffic applications, this is considered one of the major sources of energy waste. Note that carrier sensing, which a MAC protocol requires to sense the current status of the channel, is not a part of idle listening.

- **Collisions:** A collision is a wasted effort when two frames collide with each other and are discarded because the receiver has to drop the overlapped information. A collision usually results in retransmission and drains more energy in transmitting and receiving extra packets. The half duplex nature of the wireless medium precludes collision detection, thereby increasing the responsibilities of the MAC protocol. The high density of the deployed nodes, on one hand, helps improving network connectivity without compromising the transmission power. However, on the other hand, it increases collision probability for the MAC protocol by increasing the number of nodes contending for the channel. For applications that do not need a reliable link layer, i.e., retransmission of packets, collisions may have a smaller impact. However, the packet loss decreases the accuracy of the application. The several approaches proposed to avoid collisions are discussed in Section 3.1.

- **Overhearing:** An overhearing occurs on the wireless broadcast medium when the node receives and processes a gratuitous packet that is not addressed to it. In the dense network and under heavy traffic situations, this could lead to a serious problem. Nevertheless, some MAC protocols also leverage overhearing to infer important information about the channel, current transmission, or link status [27, 28, 76].

- **Control packet overhead:** An increase in the number and size of control packets results in overhead and unnecessary energy waste for WSNs, especially when only a few bytes of real data are transmitted in each message. Such control signals also decrease the channel capacity. The nodes use control packets such as RTS/CTS/ACK frames , synchronization messages, or long preambles (further discussed in Chapter 3) to cope with the hidden and exposed terminals and to reduce idle listening, overhearing, and collisions in WSNs. A balanced approach is required so that the required number of control packets can be kept at minimal.

- **Over-emitting:** An over-emitting or a deafness occurs due to the transmission of the message when the destination node is not ready to receive it.

- **Complexity:** Computationally expensive algorithms might decrease the time the node spends in the sleep mode. They might limit the processing time available for the application and other functionalities of the protocol. An overly simple MAC algorithm can save higher energy than a complex one, but it may not be able to provide the complex functions such as adaptation to traffic and topology conditions, clustering, or data aggregation.

Duty cycling, discussed later in this chapter, i.e., putting radios of sensor nodes in low power sleep mode for most of the time and waking them up shortly in between to check the channel activity is widely considered as the powerful mean to cope with most of the energy related issues in WSNs. However, it also results in high latency and low throughput as nodes remain in sleep mode most of the time. Moreover, a deep consideration is required to select a proper duty cycle value, since more frequent switching of the radio between on and off modes increases rather than decreases energy consumption for the node.

Apart from energy efficiency, several applications of WSNs may need delay bound operations. Unlike traditional distributed systems, the timeliness guarantee for WSNs is more challenging. They interact directly with the real world, where the physical events occur in an unpredictable manner with different traffic and delay requirements. Duty cycling, dynamic topology, and limited memory and computation power also restrict the design space we could trade off.

The MAC protocol can coordinate communication of nodes by waking up their radios at the scheduled times (further discussed in Chapter 3) in order to provide deterministic delay and better energy efficiency. However, these centralized schemes usually require a system wide synchronization among nodes, which causes high complexities and control overhead, long queuing delays, and serious scaling and adaptability problems in a dense WSN. To reduce energy and delay further, the MAC protocol can adapt itself with the varying network topology, size, and traffic characteristics. However, such adaptation often incurs complexity by increasing the processing and memory usage for the node. The granularity and rate of adaptation are also the factors affecting the system performance. Cross-layering, i.e., interaction with other layers may allow the MAC protocol to coordinate and limit the resources needs for an operation [14, 77–79]. For example, the effective routing information could help the MAC protocol reducing energy and time consumption or vice versa, and the MAC layer could priorities different message types described by the application layer. The strong coupling of the MAC and PHY layers enables the MAC protocol to know about the current status of radio, channel, and many other elements. While cross-layering has several positive aspects, it suffers from the confined generality and interoperability issues, particularly with resource-constrained sensor nodes [14]. Concludingly, the MAC protocol for WSNs usually trades-off among several and often contradictory factors and offers a compromised performance, mostly in favor of energy.

Last but not least, selecting a proper hardware scheme also has a lasting impact on the performance of the MAC protocol and ultimately on the whole WSN system. Support for multi-channel hardware, balancing an appropriate memory size, use of wake-up radio, and selection of the packet- or bit-based radio are the important factors [15].

2.4.3 Common MAC Approaches

There is no universal *best* MAC protocol for WSNs; the design choice mainly depends on the nature of the application [3]. The stringent design requirements of the MAC protocol for WSNs can be met by a plethora of approaches. The most widely used approaches in designing such MAC protocols, along with their implications, are outlined below.

2.4.3.1 Channel Access Methods

Acquiring and releasing the channel is the core in the design of any MAC protocol, and in a dense and energy-limited WSNs its importance increases even more. Several widely used channel access methods with their strong and weak points are elaborated throughout Chapter 3.

2.4.3.2 Duty Cycling

Though the application domain of WSNs is diverse and broad, environmental monitoring and surveillance have been the most visible applications of WSNs. The data traffic generated and processed by sensor nodes in such applications can be distinguished in two different classes; *periodic traffic* and *event-based traffic*. The periodic traffic class senses the environment usually at a regular interval and collects the information relating to one or more parameters for a physical object and report to the sink node [58, 60, 61]. In event-based traffic class, nodes do not follow the periodic monitoring mechanism but report the sink node or sound an alarm when something significant occurs in the sensor field [62, 80, 63]. Nodes in this class remain idle most of the time but usually generate a burst of packets during the short time when an event occurs.

A WSN generally generates much less data traffic and sends very small data frames as compared to traditional wired or wireless networks. Sensor nodes therefore remain idle most of the time either waiting for their periodic turn to generate data or listening the idle channel for something to occur. Since the radio consumes as much energy during idle listening as in receiving data packets, switching it into low-power sleep mode and waking up shortly at a periodic interval can significantly conserve energy of nodes. Moreover, many other nodes may fall in the transmission range of a node, it is often desirable to turn off radios in the majority of neighboring nodes, so that collisions, overhearing, and redundancy can be avoided. All these facts are sketched down in Figure 2.5, where the node periodically switches its radio between sleep and listen periods rather than constantly listens the idle channel. It turns the radio to sleep mode for a *sleep period*, t_S, and wakes it up to check the medium for a short *listen period*, t_L. The sum of the sleep period and the listen period is called a *wake-up period*, t_W, whereas the ratio of the listen period to the wake-up period, t_L/t_W, is called *duty cycle* of the node.

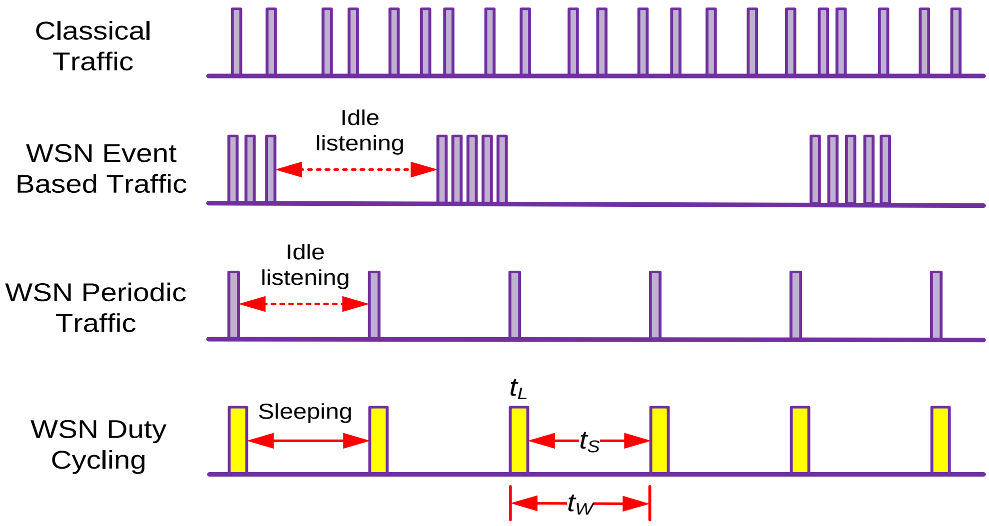


Figure 2.5: Duty cycling in WSNs. Nodes usually generate/process data at a very low rate and use the channel occasionally, either periodically or whenever an important event occurs. Therefore, in order to reduce idle listening and overhearing, nodes perform duty cycling, remain in the sleep mode most of the time, and wake up shortly to sense the channel.

Duty cycling significantly increases system lifetime of a dense WSN by reducing idle listening and overhearing among nodes. Importantly, nodes usually do not need any additional hardware or a complex algorithm to perform duty cycling. However, these advantages have other implications too. The transceiver is usually kept in the sleep mode most of the time, which could end up in a significant competition among neighbors at wake-up periods. This could ultimately lead to collisions, low throughput, and high network latency for a WSN, particularly in heavy load situations. The important question that arises here is to select an optimal value of duty cycle for an application. Choosing a long sleep period induces significant per-hop latency, since a sending node has to wait an average of half a sleep period before the receiver can accept packets. Too short sleep phases, i.e., more frequent switching of the radio between on and off modes also outweighs the benefits of duty cycling because the switching is not instantaneous and consumes additional energy [1]. Hence, the optimal selection of the duty cycle value is a critical step towards achieving the desired system performance.

2.4.3.3 Timeliness

While designing the MAC protocol for WSNs, the timeliness factor is often ignored by researchers. With ever increasing applications of WSNs in many diverse fields, new concepts of offering timeliness related QoS are inevitable. Generally, timeliness related applications can be categorized into Hard Real-Time (HRT) and Soft Real-Time (SRT) based applications [81]. A deterministic end-to-end delay is required in HRT applications, where a strict deadline is applied on the arrival of messages. Alternatively, a tolerable and probabilistic delay guarantee is supported for SRT applications.

Limited resources, low node reliability, dynamic network topology, and direct interaction with the physical world makes HRT very difficult in WSNs. With a time scheduling mechanism, which is discussed in Chapter 3, a bounded and predictable delay in WSNs can be achieved [30–32]. However, even in that case, along with other implications, the average queuing and access delays are much higher, since a node has to wait for its allocated slot before accessing the medium. As a result, the probabilistic based SRT guarantee in many WSN related applications is mostly permissible.

2.4.3.4 Topology Control

The goal of topology control is to build a reduced topology by dynamically changing transmitting range of nodes in order to save energy and preserve important network characteristics, such as connectivity and coverage [82]. Since the transmission energy often dominates the total communication energy and grows with the increase of transmission distance, topology control can reduce this consumption by forcing packets to travel through multiple hops. The topology control function is usually located between the MAC and network layers and interacts with both of them. Topology construction can also be exercised by dynamically turning unnecessary nodes off or by creating hierarchies in the network, so that specific nodes can be directed to take over certain coordination functions [3]. The topology control mechanism reduces energy consumption by reducing collisions, contentions, and exposed terminal problems. However, idle listening, overall latency, complexity, and increased probability of packet loss remain core issues with this mechanism.

Figure 2.6 illustrates that energy consumption can be reduced by minimizing transmission range of a node, so that packets can only travel through multiple hops rather than a single long hop [72]. A linear network is considered in this example where n nodes are equally spaced by distance d. If node 1 directly communicates with node n over the total distance of D, and if path loss exponent of 2 is assumed, then the received power at a distance D can be given by the Friis free-space propagation model, shown in (2.1).

$$P_r = P_t \times G_t \times G_r \times \left(\frac{\lambda}{4 \times \pi \times D}\right)^2 \tag{2.1}$$

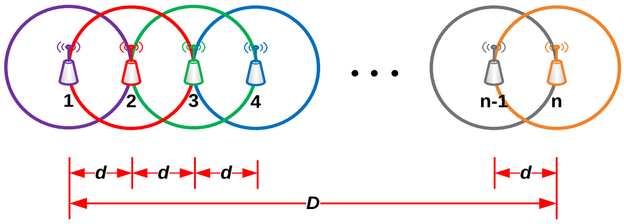


Figure 2.6: A simple topology control example in WSNs, where nodes dynamically change their transmission power to save energy. They transmit packets via their adjacent neighbors rather than sending over the long distance.

Where P_t is the power at which the signal was transmitted, G_t and G_r are the antenna gains of the transmitter and receiver respectively, and λ is the wavelength. This equation can be rewritten as:

$$P_r = C \times \frac{P_t}{D^2} \qquad C = \frac{G_t \times G_r \times \lambda^2}{(4 \times \pi)^2} \tag{2.2}$$

Where C is a transceiver-dependent constant. If we assume $C = 1$, then the required transmission energy to transmit the signal at a distance D for the node 1 can be given by (2.3).

$$P_t = P_r \times D^2 \tag{2.3}$$

Alternatively, if the node 1 chooses to transmit a packet to the node n via its adjacent neighbors, i.e, over $n - 1 = h$ hops then the energy needed by each node to reach its immediate neighbor can be given as (2.4).

$$P_t = P_r \times d^2 \tag{2.4}$$

Therefore, the power saving with multi-hop communication is:

$$P_s = \frac{P_r \times D^2}{h \times P_r \times d^2} = \frac{P_r \times (h \times d)^2}{h \times P_r \times d^2} = h \tag{2.5}$$

Equation (2.5) concludes that the larger the number of hops h, the higher the power saving for a node. However, this is a very simplistic example that considers only the transmission energy of nodes. Factors like processing energy (with multi-hop communication $(n - 1)$ nodes process and hence consume energy as opposite to the only two nodes with single-hop communication), BER, and probability of packet loss with the varying transmission distance also need to be considered

2.4.3.5 Scheduling and Synchronization

Many WSN MAC protocols assume that nodes are time-synchronized and follow a fixed schedule to switch radios between wake-up and sleep modes [27–29]. Similarly, schemes used in [30–32] assume that synchronized nodes can only access their allocated time slots without the need of any contention, so that collision, idle listening, and overhearing can be reduced. However, in reality such time synchronization in dynamic and resource-limited WSNs is very difficult to achieve as it induces a lot of overhead and may need extra hardware. Collisions and retransmissions increase dramatically if all nodes wake up simultaneously, and network latency increases with the slot allocation schemes. Therefore, it is wise to use random and non-synchronized wake-up and sleep schedules for sensor nodes.

2.4.3.6 Cross-layering

Most of the proposed MAC protocols for WSNs follow the traditional layered architecture, where they try to improve performance only at the respective layer [27, 28, 30, 31, 45–47]. With very limited resources available to sensor nodes, a trend of the cross-layer design is emerging in order to achieve aggregate optimization among different layers. Unlike layered networks, WSNs cannot afford an isolated layer architecture due to their limited energy, storage, and processing capabilities. Moreover, application-aware communication and low-power radio considerations motivate for cross-layer architecture for WSNs. Recent studies in [77, 78, 83, 84, 79] affirm improvement in WSN performance by using cross-layering. There is, however, still much to be done to provide unified cross-layer communication architecture for dynamic and resource-constrained WSNs.

2.4.3.7 Miscellaneous Techniques

Along with common methods of the MAC designing for WSNs, some unconventional approaches have also been endeavored in the literature. Some researchers counsel for having two different channels with each node; the *data channel* and *control channel* [85]. The data channel is always kept in sleep mode except when transmission of data and/or ACK packets occurs, whereas the control channel is used to exchange control packets. This approach does not need synchronization but increases complexity in terms of hardware, cost, energy consumption, and handling of two transceivers at each node.

Other researcher opt for the positive aspects of more than one channel accessing approaches to form a common set of two or more characteristics of different methods [86–88]. Such *hybrid schemes* may improve overall performance of WSNs, but they also increase overhead and complexity as more than one channel accessing methods need to be handled simultaneously.

The protocols proposed in [89, 90] suggest shifting transmission initiation from the sender to receiver side. When the receiver is awake and ready to receive a frame, it sends a beacon and starts monitoring the channel for incoming frames for a while. On reception of the beacon, a sender node can send the actual payload. Such schemes are in fact similar to the classical ones with only the difference of who is going to start the communication. Therefore, they carry all the common problems of this domain in one or the other way. Moreover, such protocols cannot be used for broadcast and multicast communication and can increase latency and idle listening at the sender side.

Most of the referenced approaches and protocols in this chapter are detailed in the next chapter, where we present state of the art related to the MAC designing for WSNs.

2.5 Conclusion

This chapter has presented a comprehensive introduction to the design of WSN MAC protocols. In order to understand the necessary background of the WSN domain, the chapter has been commenced by explaining the architecture, applications, design objectives, and challenges of WSNs. Later on, an overall understanding of the design aspects of their MAC protocols has been presented, where we have discussed why traditional MAC protocols are not compatible with WSNs. We have outlined a set of important attributes to further building up the contents of this thesis by highlighting the basic working, services, challenges, and common approaches in designing a WSN MAC protocol.

In the next chapter, we cover state of the art of MAC protocols that have been particularly proposed for WSNs.

Chapter 3

Overview of State of the Art

No one wants to learn from mistakes, but
we cannot learn enough from successes
to go beyond the state of the art.
–Henry Petroski

THIS chapter presents a broad overview of state of the art of MAC protocols developed for WSNs. Before categorizing them in different channel accessing classes, we first explicate the brief history of channel accessing schemes. We believe that this background holds the key, since roots of many of the WSN MAC protocols are found in these conventional schemes.

3.1 Channel Accessing Chronology

The collision prone nature of the wireless broadcast medium requires an efficient channel accessing method to arbitrate access to the shard medium so that collision-free communication among nodes can be offered. Broadly speaking, *contention based* and *contention free* are the two major categories of accessing the channel [2, 57]. In contention based networks, devices contend with each other to gain access of the channel. Contention free networks schedule the channel, either in time or frequency, where devices can only access their allocated channel slots, and thus communicate with the central node in a collision free manner. We next go through the background of the contention based protocols, the advantages and disadvantages of contention free or scheduling based protocols are briefed in Section 3.2.2.

The attempt to find the answer of *who and how to access the channel* has been started long ago. The *Additive Link On-Line Hawaii System (ALOHA)* protocol [73, 74], also referred to as *pure ALOHA* and proposed in 1970s, is one of the pioneer protocol of this category. It allows devices to transmit as soon as they have data to send. After sending a packet, the sender expects an ACK from the receiver within a time period. If no ACK is received, ALOHA considers that the packet has been lost in a collision. ALOHA addresses the collision issue by forcing the sender to wait for a random time before the next trial. Though ALOHA is a simple and decentralized MAC protocol and works well under low loads, its maximum channel utilization is only 18.4% [74]. This utilization can be doubled with the *slotted-ALOHA*, where the time is subdivided into slots. The node is allowed to start a transmission only at the beginning of a slot, so that collision can occur only at the beginning of the slot. Though slotted-ALOHA reduces the collision probability, it needs synchronization among nodes.

The CSMA approach is widely used in both wired and wireless LANs. To avoid collision, CSMA introduces the capability of sensing the transmission of other nodes before a node starts with its own transmission. In order to decide what to do in case if the node finds the carrier busy, CSMA uses two major versions: *non-persistent CSMA* and *persistent CSMA* [3, 7]. In non-persistent CSMA, if the node finds the channel busy, it waits for a random time and then starts listening the channel again. Persistent CSMA has many variants that use different *backoff algorithms* to decide the behavior of the node in case of the busy medium. For example, with *1-persistent CSMA*, the node continues to listen until the channel becomes idle. In *p-persistent CSMA*, if the node finds the channel to be idle, it transmits with probability p. It can postpone its transmission with probability $q = 1 - p$, wait until the next slot, and transmit again with probability p. With 1-persistent CSMA, blocked devices can queue up and lead to significant collisions once they find the channel idle, whereas non-persistent and p-persistent avoid collisions at the cost of higher delays.

A *contention window* based backoff algorithm [1] is shown in Figure 3.1, where the node first listens to the channel for an IFS time. If the channel is idle during this period, the node may transmit immediately. Otherwise, it defers its transmission and continues to monitor the channel until the current transmission is over. To prevent a simultaneous access of the channel in case of more than one node defer their transmission and sense the channel idle, the node waits for another IFS period once the current transmission is over. If the medium remains idle for this period, the node picks up and starts a timer with a random value from the current contention window. The timer is decremented after each slot. The first node with its clock reaching to zero starts transmission. Other nodes sense this transmission and freeze their backoff clocks, to be restarted after completion of the current transmission in the next contention period. In case of a collision or missing ACK, the size of contention window can be increased according to a binary exponential backoff scheme.

Though CSMA protocols increase the throughput efficiency of ALOHA protocols, they are still susceptible to collisions and less throughput, especially via the famous *hidden and exposed terminal problems* [3, 6]. Since the sender cannot detect interference at the receiver end, the hidden terminal problem may stem when more than one node, which cannot hear each other but the receiver node only, start transmitting a packet. Alternatively, the exposed terminal problem

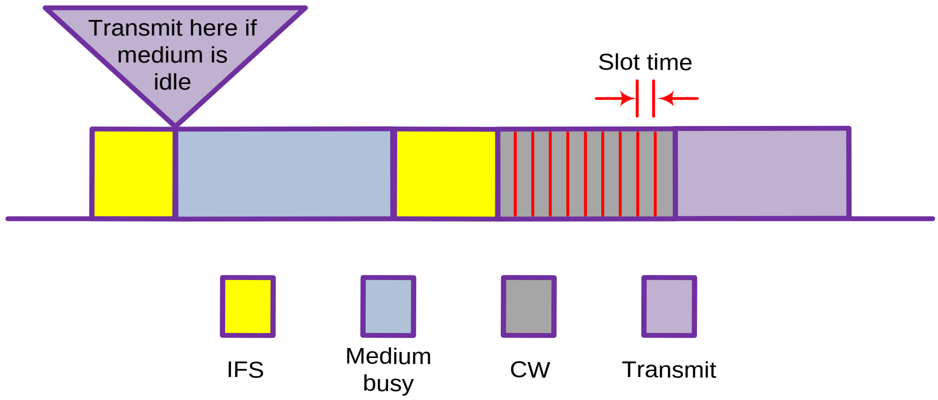


Figure 3.1: A simple CSMA scheme working on the contention window based backoff algorithm, where nodes first listen the channel for an IFS period. If the channel is not idle after an IFS, nodes perform backoff.

may occur when the node is prevented to communicate with its neighbors due to a neighboring transmitter. The node senses the carrier busy and defers the transmission when another neighboring transmitter, which is in the range of first one, is transmitting, even if their intended receivers are out of range of each other and are not causing any collision.

The *CSMA-CA* scheme, where CA stands for Collision Avoidance, is among other approaches introduced to minimize the impact of the hidden and exposed terminal problems. CSMA-CA introduces a four-way handshake mechanism between the sender and receiver for successful communication between them. Figure 3.2 shows the CSMA-CA algorithm used by the DCF of IEEE 802.11 and many other WLANs [1]. A sender first senses the medium for an IFS period of the DCF, referred to as DIFS. If the medium is found to be idle, the sender initializes the handshake by sending a short RTS frame to the intended receiver. The receiver, on finding the medium idle for a SIFS time period, responds with a CTS frame. Once the sender receives the CTS frame, it waits for a SIFS period and then sends the data packet. The receiver can then send the ACK packet to confirm the successful reception of the data packet. Since $SIFS < IFS$, this handshake has a higher priority over other transmission attempts. Other nodes can defer their transmission by using a backoff algorithm till the channel becomes idle.

With IEEE 802.11, the sender adds the duration of the upcoming data packets in the RTS frame, and the receiver also copies same duration in the CTS frame. Any other node that tries to transmit during this handshake period and sense any of these four packets extracts this duration and puts it into the NAV table. This time is decreased by each time slot and a node can perform carrier sensing once its NAV expires. Hence, NAV helps a node knowing the remaining time of the current transmission and freeing it from the continuous channel sensing. Though CSMA-CA provides substantial improvement over the hidden and exposed node problems, it cannot completely eliminate collisions. The four-way handshake scheme also

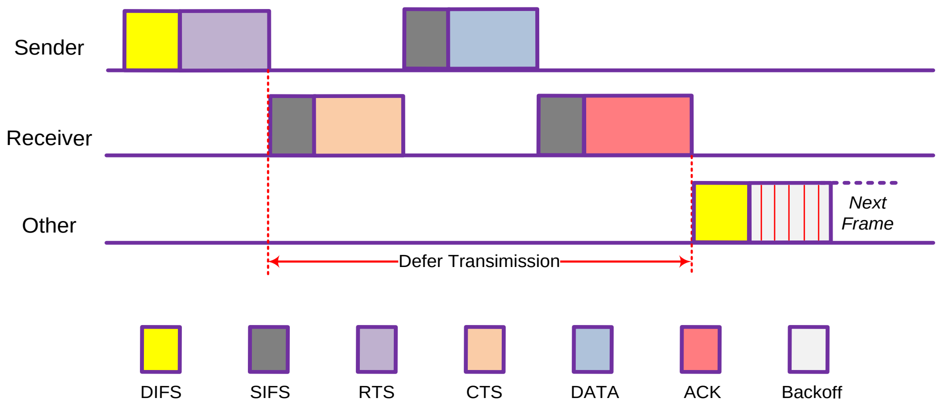


Figure 3.2: IEEE 802.11 DCF based CSMA-CA protocol with the ACK mechanism, where neighbors sense the duration of the current transmission by overhearing RTS or CTS frames.

incurs additional overhead in transmitting extra control packets.

The *Multiple Access with Collision Avoidance (MACA)* protocol [91] aims to reduce the hidden and exposed terminal problems of CSMA-CA. It does not use carrier sensing for devices but adds the remaining data exchange length to the RTS and CTS messages. On overhearing such messages, a device can determine how long to delay before attempting a transmission on an 'idle' channel. MACA also allows a device that overhears an RTS frame but does not receive/overhear the expected CTS frame, to start with its transmission. This improves the exposed terminal problem of CSMA-CA, where a device that overhears an RTS always remains quiet. Note that MACA uses the RTS and CTS control frames but not the ACK.

The *MACA for Wireless LANs (MACAW)* protocol [76] is an extension to MACA, which enables nodes to send an ACK after successfully receiving a data packet to ensure reliability. If the sender does not receive the ACK even after sending the data packet, it sends the RTS for the retransmission, but the receiver returns the associated ACK instead of a CTS. MACAW also adds an extra control frame called *Data Sending (DS)* between the CTS and data messages that confirms devices near the source that the current transmission will last for at least the entire data packet. If the device overhears an RTS but not a DS, then it knows that another transmission may occur, since the current destination did not transmit a CTS.

MACA tries to improve the hidden and exposed terminal problems and enables nodes know the length of current transmission and hence delay for the optimal time, whereas MACAW provides data reliability at the MAC layer. However, like CSMA, these schemes require devices to constantly sense the channel. MACAW uses an extra control packet, which manifolds the pre-existing control overhead to these schemes.

Having gone through with the basic medium access mechanisms, we now turn our focus on the MAC approaches and protocols that have specifically been designed for WSNs.

3.2 Classification of WSN MAC Protocols

In order to meet the stringent design requirements of WSNs, several MAC protocols have been proposed in the literature. These protocols depending on how they allow nodes to access the channel can be classified into the general categories of *contention based*, *scheduling based*, *channel polling based*, and *hybrid protocols* [39]. Figure 3.3 depicts this classification.

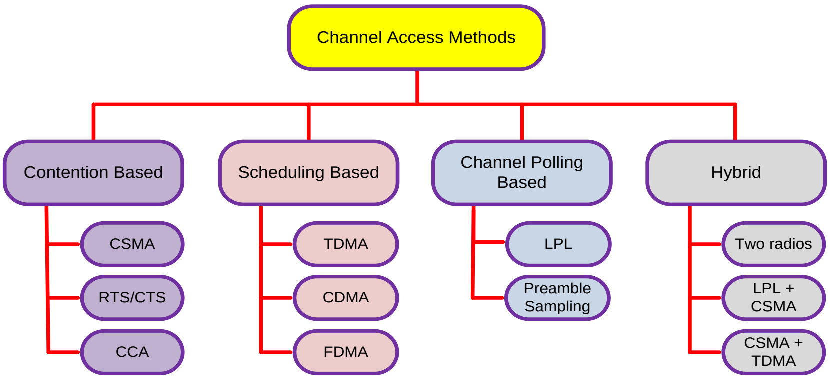


Figure 3.3: Channel accessing taxonomy in WSNs. The four major categories can be further divided into the subcategories depending on the accessing or naming scheme they offer.

3.2.1 Contention Based MAC Protocols

As discussed, nodes using contention based schemes compete with its neighbors to acquire the channel. Before any transmission, a node first senses the carrier. If the carrier is found to be idle, the node starts with its transmission, otherwise it defers the transmission for some random time usually determined by a backoff algorithm. Contention based MAC protocols consume less processing resources and are suitable for event-driven WSN applications. They are flexible to network scales and dynamics as no clustering and/or topology information is required for their working. Each node can independently decide for contention without the need of control frame exchanges. However, transmission with this approach is purely handled by the sender, and in result the problems of hidden and exposed terminals may occur causing collisions, overhearing, idle listening, and less throughput.

Many MAC protocols falling in this category consider that the contention times of nodes are synchronized according to a schedule, i.e., at each periodic interval, all neighboring nodes wake up simultaneously to exchange packets [27–29]. In that case, chances of collisions are very high, since all neighboring nodes compete for the channel simultaneously. The collided packets are usually retransmitted, which results in higher energy consumption and delays. The formation and maintenance of synchronization in resource-limited WSNs lead to complexity and communication overhead and may require special hardware and/or algorithms. Moreover, the clock drift on each node can affect the schedule coordination and synchronization by causing timing errors [57].

An ample amount of WSN MAC protocols working on the contention based mechanism is available in literature. We depict next some of the representative protocols of this category.

3.2.1.1 S-MAC

The design of the Sensor-MAC (S-MAC) [27] was one of the initial attempts to significantly reduce idle listening, collisions, and overhearing in WSNs by putting nodes in *listen* and *sleep* periods. Listen periods are normally of fixed size according to some PHY and MAC layer parameters, whereas length of sleep periods depends on the predefined application based duty cycle parameter. S-MAC attempts to coordinate neighboring nodes by letting them share common listen periods according to a *schedule*. This requires formation and maintenance of synchronization among nodes.

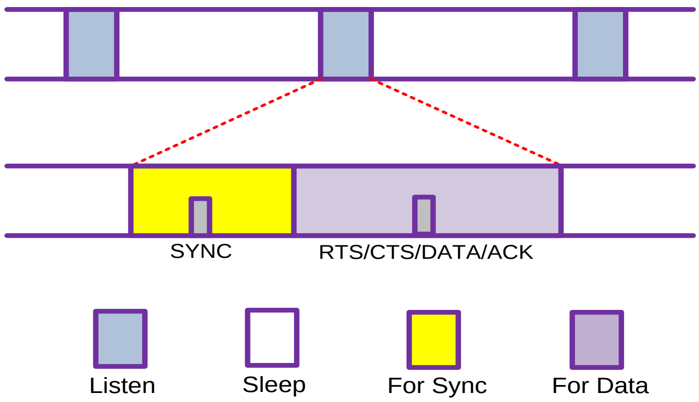


Figure 3.4: Periodic listen-sleep schedule with S-MAC. Nodes using S-MAC perform duty cycle and switch their radios between sleep and listen periods. A listen period is used to exchange sync as well as data/control packets.

The listen period is further divided into two intervals; the *SYNC period* and *Data period*, as drawn in Figure 3.4. During the SYNC period, a node tries to receive a SYNC packet from its neighbors, which consists of the sender ID and the remaining time until the sender switches to sleep mode. At initialization, if the node does not hear the schedule from its neighbors, it immediately chooses its own schedule and broadcasts it. This node is called the *synchronizer*, and other nodes begin to synchronize themselves with it. Alternatively, if the node finds a schedule, it stores the schedule in its schedule table, adapts it, and thus becomes a *follower* node. The data period is used to exchange data related messages, which may include RTS, CTS, DATA, or ACK messages. In order to minimize the hidden terminal effect, S-MAC uses the RTS/CTS handshake scheme for unicast packets. Broadcast packets are sent without using the RTS/CTS handshake. To minimize costly retransmissions, S-MAC provides a *message passing* mechanism, where long messages are fragmented into short frames and sent in a burst. The RTS/CTS is only required before transmitting the first short frame.

With S-MAC, a node may receive multiple schedules from its neighbors. In that case, it adapts to all schedules and acts as a *border node* for two or more *virtual clusters*. Thus a S-MAC network is most likely to have several virtual clusters. Figure 3.5 shows an example of the virtual cluster formation with S-MAC. Border nodes need to wake up at the listen intervals of all these schedules for the successful formation of virtual clusters. Since a node tries to follow exiting schedules during the initialization period, S-MAC expects that the node adapts multiple schedule rarely.

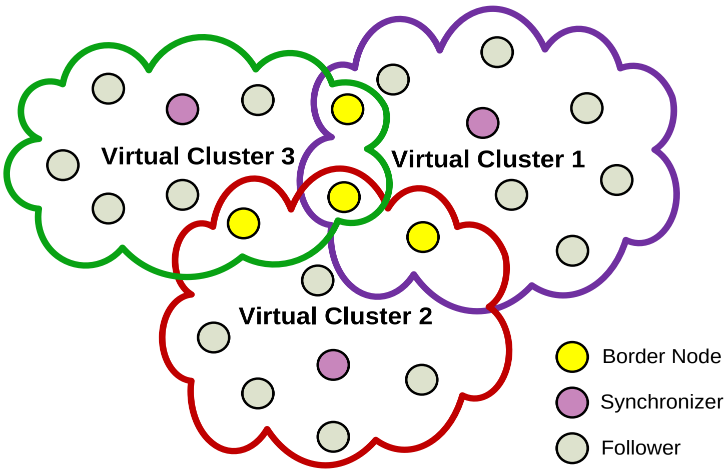


Figure 3.5: Virtual Cluster Formation with S-MAC [1]. S-MAC forms a virtual cluster containing a set of nodes synchronized with each other. Since the whole network cannot be synchronized together, border nodes are allowed to have different schedules. This enables inter-cluster communication and some sort of adaptivity with S-MAC.

Although S-MAC reduces idle listening and overhearing for duty cycled nodes, it has several other drawbacks. S-MAC is rigid and optimized for a predefined set of workloads, since there is no mean to adapt the length of listen and sleep periods with changing traffic conditions. As discussed earlier, the formation, maintenance, and compliance of synchronization has serious consequences in WSNs. Longer and fixed sleep periods of S-MAC have stark impact on system latency. It could be worsen if intermediate nodes on a route do not share a common schedule. To decrease latency, an *adaptive listening* mechanism is employed with S-MAC. A node x, that overhears the on-going transmission of its neighbors, receives the RTS/CTS packet, extracts the remaining time of the transmission from the duration field of the RTS/ CTS packet, and wakes up just at the end of the transmission. The node receives and processes the packet if the packet is destined for it, otherwise it goes back to sleep mode. However, once the node x receives the packet, it has to wait for the next listen period to find the next hop and forward the packet further, which limits the advantages of the adaptive listening. Energy consumption may be increased with the adaptive listening, since all the overhearing nodes need to wake up at the end of current transmission.

Border nodes expend more energy than non-border nodes as they wake up at each schedule they follow in order to maintain network connectivity among clusters. S-MAC does not provide a mean to control the size of virtual clusters and the number of border nodes. With fragmentation in S-MAC, overhead and retransmission can be reduced, it, however, comes at the expense of unfairness, since the node reserves the channel for a whole burst duration. A neighboring node carrying delay bound data would have to wait longer to gain access of the channel.

Several improvements and enhancements have been proposed to overcome the weaknesses of S-MAC, here we discuss some of the popular variants of S-MAC. Most of them, by default, inherit overhead and complexity incurred in synchronizing common listen and sleep schedules for nodes.

3.2.1.2 T-MAC

The Time-out MAC (T-MAC) [28] protocol improves energy efficiency of S-MAC protocol, especially under variable traffic conditions, by adaptively shorten listen periods of sensor nodes. Unlike S-MAC, where listen periods are always rigid, the listen period of a T-MAC node ends when no activation event has occurred for a threshold period *TA*. This dynamic aspect frees the application from selecting an appropriate duty cycle value. The comparison between S-MAC and T-MAC shown in Figure 3.6 confirms this improvement. However, this improvement is heavily dependent on the *TA* value. *TA* must be long enough so that a node can sense the carrier and hear a potential CTS from a neighbor.

Although T-MAC performs better under variable loads, synchronization of the listen periods within virtual clusters may partially break down. This could lead to the *early sleep* problem for T-MAC nodes. The early sleep happens when a node, especially third hop one, goes to the sleep mode when a neighbor still has messages for it. However with T-MAC, Future Request To Send (FRTS) frames can be sent to the third hop nodes either to extend their TA expiration, or to let them awake by the appropriate time. T-MAC saves more power than S-MAC and minimizes collisions and redundancy, since idle nodes switch back to the sleep mode relatively earlier. However, this comes at the cost of reduced throughput and higher network latency. T-MAC also suffers from synchronization and scaling problems.

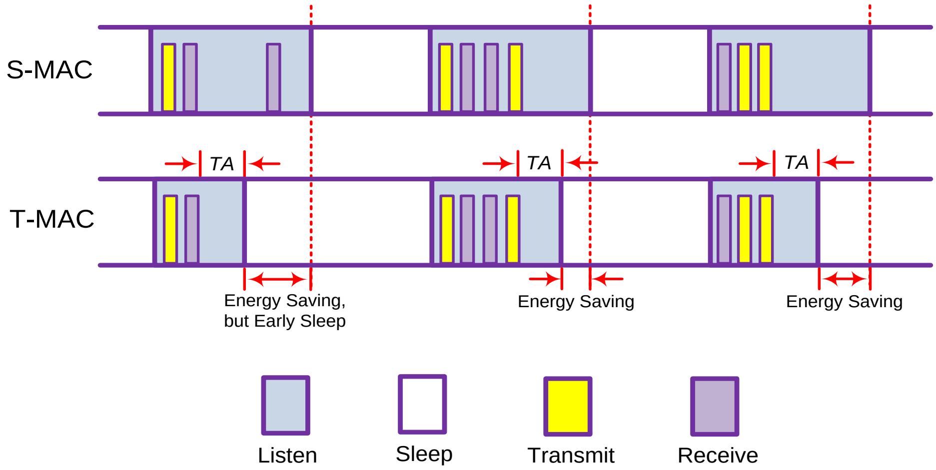


Figure 3.6: T-MAC vs. S-MAC. T-MAC reduces energy consumption of S-MAC by adaptively terminating the listen period of the node if no activation event has occurred for the TA period. However, this early sleeping could result in packet loss, low throughput, and high network latency.

3.2.1.3 DSMAC

The Dynamic S-MAC (DSMAC) [29] improves latency of S-MAC by dynamically adjusting the duty cycle value of nodes as per traffic and energy conditions. A node using DSMAC keeps track of its energy consumption level and average latency it has experienced and tries to dynamically adjust its duty cycle accordingly. All nodes start with the same duty cycle value and share their one-hop latency values in the SYNC period. When the node notices that the traffic has been increased or low latency is required, it adds extra active periods, shortens its sleep time, and sends an updated SYNC message to its neighbors. On receiving the updated SYNC message, a neighboring node checks its queue for packets destined to that node. If there is a one, it doubles its duty cycle provided that its battery level is above a specified threshold.

Latency observed with DSMAC is better than S-MAC, but nodes consume higher average energy and use less frame duration. Thus, DSMAC achieves less throughput at high traffic. With varying duty cycle values, synchronization of a virtual cluster may get affected. A high duty cycled node can receive a SYNC message sent by a node operating at a low duty cycle. Thus, complexity in adapting duty cycle values, particularly within a virtual cluster and under high traffic loads, manifolds the pre-existing synchronization overhead.

3.2.1.4 Global Schedule and Fast Path Algorithms

As mentioned earlier, nodes with S-MAC protocol may follow more than one schedule and consume higher energy by spending more time in active periods. To minimize the number of active schedules, the number of border nodes, and latency of

S-MAC, two algorithms have been proposed [92]. The first algorithm called Global Schedule Algorithm (GSA) minimizes the number of active schedules of S-MAC. GSA uses age of the schedule to determine which schedule to prefer and keep. On the availability of more than one schedules for a node to choose with, it prefers the oldest schedule. Over the time, all nodes migrate toward the oldest common global schedule in the network.

The second algorithm is called Fast Path Algorithm (FPA) that provides fast data forwarding paths by adding additional wake-up periods on the nodes along paths from sources to the sink node. Given a source, sink, and the path between them, additional wake-up periods along the path are added such that they occur exactly when the previous-hop node is ready to send the packet.

Results achieved in [92] show that more than half of nodes using S-MAC have more than one active schedules, and GSA converges to one schedule in a network of 50 Mica2Dot nodes deployed in a linear fashion quite well. However, in addition to the synchronization overhead, nodes using GSA and FPA need more processing in deciding the fate of schedules and data forwarding paths.

3.2.1.5 STEM

The Sparse Topology and Energy Management (STEM) protocol [85] uses two different channels for each node; the *wake-up channel* and *data channel*, as shown in Figure 3.7. The wake-up channel only monitors for control signals, whereas the data channel is solely used for data packets and remains in sleep mode most of the time. On the wake-up channel, the time is divided into fixed-length *wake-up periods*, which are further subdivided into *listen* and *sleep* periods.

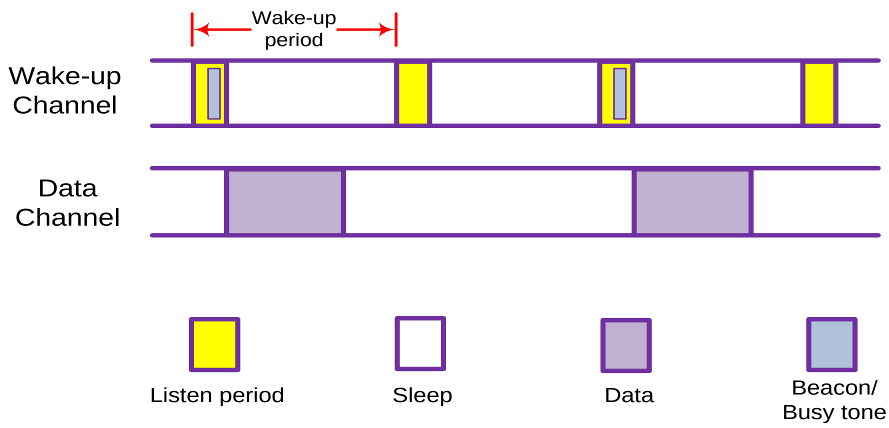


Figure 3.7: The working of STEM. A sender sends a beacon on the wake-up channel with the STEM-B and a busy tone with the STEM-T variant to pull in the intended receiver. A node turns to the data channel only if a beacon or a busy tone is found on the wake-up channel.

To gain the attention of the receiver, two different variants of STEM are used. In **STEM-B**, the transmitter tries to wake up the receiver by sending contention-free *beacons* on the wake-up channel, each containing source and destination addresses. The sender sends these beacons at least for a complete wake-up period. However, it stops sending further beacons as soon as it receives an ACK frame from the receiver. In that case, both the sender and receiver switch on their transceiver for the data channel and start communicating. The non-targeted nodes can go back to sleep mode once they recognize that the packet is not addressed to them. In **STEM-T**, the transmitter sends a *busy tone* signal on the wake-up channel, but it contains no destination address, and in result all the neighbors may sense it and switch on their data channels. STEM-T looks very similar to the traditional channel polling based protocols except of using two separate channels instead of one.

STEM-B cuts back the number of beacons to be sent as the transmitter does not always need to send a beacon burst of full wake-up period length. However, more than one transmitter might send the beacon simultaneously, causing a beacon collision. STEM-T uses a simpler and cheaper transceiver on the wake-up channel, but busy tones are sent for the maximum time. Non-targeted neighbors can receive busy tones and unnecessarily switch on transceivers for their data channels. On a whole, the employment of two different channels for each node is a complex and expensive process in itself.

3.2.2 Scheduling Based MAC Protocols

Scheduling based schemes assign collision-free links to each node in a neighborhood usually during the initialization phase. Links may be assigned as TDMA slots, FDMA bands, or CDMA based spread spectrum codes. However, due to the complexities incurred with FDMA and CDMA schemes, TDMA schemes are preferred as scheduling methods for WSNs. With TDMA schemes, the system time is divided into slots, which are then allocated to all the nodes in the neighborhood. A *schedule* regulates which participant may use which resource at what time. The schedule can be fixed or computed on demand (or a hybrid) and is typically regulated by a central authority. A node can only access its allocated time slot and does not need any contention with its neighbors.

The cardinal advantages of scheduling based schemes include minimum collisions, less overhearing, and implicitly avoidance of idle listening. They also provide a bounded and predictable end-to-end delay. However, the average queuing delay is much higher, since the node has to wait for its allocated slot before accessing the channel. Overhead and extra traffic required in setting up and maintaining synchronization among nodes, lacking of adaptability to varying traffic and topology conditions, reduced scalability, and low throughput are the other major concerns with these schemes. The performance of a network is highly subjected to the degree and oftenness to which it undergoes organization and reorganization. Allocating conflict-free TDMA schedules is indeed a difficult task in itself. A peer-to-peer based communication is also not possible with scheduling based schemes as nodes are normally allowed to communicate only with the central authority.

In order to avoid costly global state distribution or synchronization, several scheduling based MAC protocols adapt *clustering*, where nodes usually need to share information locally with a *cluster head* [30, 93]. Clustering can ease the scaling problem by considering the cluster as a single entity. It can also save energy by distinguishing local traffic with the global one. However, nodes in a cluster need to communicate many coordination messages to maintain the cluster. The formation and adaptation of cluster with network dynamics and the selection and rotation of the cluster head to evenly distribute energy consumption can be the complex issues for lightweight WSNs. The range of a cluster, responsibilities of the cluster head, and the frequency and extent of the cluster reformation are also the critical factors [14].

3.2.2.1 LEACH

The Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol [30] is mostly a scheduled based protocol, which divides a dense and homogeneous WSN into several clusters. Each cluster is supervised by a cluster head, which is responsible for creating and maintaining TDMA schedules, communicating with its cluster members, and forwarding received messages to the sink node. As the cluster head is always switched on, the chances of a cluster head to die earlier are high. However, LEACH uses a randomized rotation mechanism for selecting a cluster head. Each node can independently decide to become the cluster head with preference is given to the node that has not been cluster head for a long time. Thus LEACH tries to distribute the energy among nodes in an evenly manner.

LEACH works in *rounds*, each further divided into the *set-up* and *steady-state* phases. The cluster formation occurs during the set-up phase, where each cluster head broadcasts an advertisement (ADV) message by using CSMA to invite its members. The cluster head then creates and broadcasts a TDMA schedule for nodes that have sent join-request (REQ) to it. In order to reduce inter-cluster interference, the cluster head chooses a random CDMA code for its members. Once the set-up phase is completed, the steady-state phase starts, where a node can transmit data to its head by using the allocated slot. Upon receiving packets from its members, the cluster head aggregates and sends them to the sink node. Since LEACH does not allow any inter-cluster communication, the cluster head directly communicates with the sink node using CSMA.

The cluster head has to perform highly computational and energy consuming tasks. It prepares and maintains the TDMA schedule, remains awake for the whole round, aggregates data, and transmits it directly to the sink node. LEACH guarantees that each member node belongs to at most one cluster. However, due to an ADV collision, it cannot guarantee that each member node belongs to the cluster. In that case, LEACH considers that all nodes are within the range of the sink node. The lack of such multi-hop communication capabilities severely limits the network scalability of LEACH.

Moreover, the channel under-utilization occurs with LEACH as it considers that nodes always have data to send during their allotted time. Perfect correlation among nodes is assumed, which is hardly possible in WSNs.

3.2.2.2 TRAMA

The TRaffic-Adaptive Medium Access (TRAMA) protocol [31] is mostly a TDMA based protocol. It creates schedules for time-synchronized nodes in a distributive manner based on traffic information. The system time is divided into *cycles*, each containing the *random access* and *scheduled access* periods. The random access period consists of a collection of signalling slots, whereas the scheduled access period contains a collection of data transmission slots as depicted in Figure 3.8.

TRAMA consists of three components: the *Neighbor Protocol* (NP), *Schedule Exchange Protocol* (SEP), and *Adaptive Election Algorithm* (AEA). The NP works during the random access period and is used to exchange one-hop neighbor information and to gather two-hop topology information for each node. During the SEP, the node transmits its current transmission schedule and also picks up schedules of its neighbors. To win a time slot, the node computes its own priority and the priority of all its two-hop neighbors for each time slot. The AEA also works during the scheduled access and uses neighborhood and schedule information to select the transmitters and receivers for the current time slot, letting all other nodes to switch to the sleep mode. For efficient channel utilization, the AEA uses traffic based information that is exchanged among nodes during SEP. It also attempts to reuse slots that are not used by the selected transmitter.

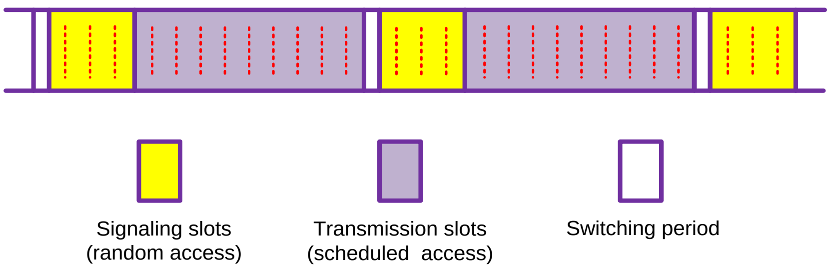


Figure 3.8: Time slot organization of TRAMA. During the random access period, nodes perform contention-based channel acquisition, and one-hop neighbor information is propagated by using signalling slots. Transmission slots are used for collision-free data exchange and schedule propagation among neighboring nodes.

A node using TRAMA can have one of the three states: transmit, receive, and sleep. The state of the node is determined based on its two-hop neighborhood information and the schedules announced by its one-hop neighbors. The node switches to transmit state if it has data to send with the highest priority among its contending set. If not, the node consults the schedule sent by the current transmitter. If the transmitter has traffic destined to this node in the current slot, it stays in the receive mode, otherwise it goes to the sleep mode.

Reuse of time slots, utilization of neighborhood and traffic information, and hybrid scheme are the positive features of TRAMA. Simulation results presented in [31] show higher percentage of sleep time, less collision probability, and better data delivery with TRAMA as compared to S-MAC and IEEE 802.11. However, all nodes in TRAMA are defined to be either in receive or transmit states to exchange schedules during the random access period. And for each time slot, every node calculates priorities for itself as well as for its two-hop neighbors. This results in significant computation, large queuing delays, and ineffective channel and memory utilization due to the fact that the two-hop neighborhood in a dense WSN could be reasonably large.

3.2.2.3 SMACS

The Self-organizing Medium Access Control for Sensor networks (SMACS) protocol [32] is a distributive and infrastructure building protocol, which forms a flat topology for WSNs. Nodes using SMACS discover their neighbors and establish transmission/reception schedules for communicating with them without the need of global synchronization or clustering. SMACS combines the neighbor discovery phase with the channel assignment phase. It assigns the channel to a link immediately after the existence of the link is discovered, instead of waiting to finish the network-wide neighbor discovery process.

To reduce collisions between adjacent links, each link operates on a random FDMA or CDMA code. Each node regularly executes the neighborhood discovery procedure and establishes the *directional* link to each discovered neighbor by assigning a time slot to that link. A node maintains a TDMA-like *superframe* to communicate with known neighbors. The fixed-size superframe is further divided into smaller and variable size frames. It helps the node maintaining its time slot schedules with all its neighbors, such that nodes are required to direct their radios to the proper FDMA or CDMA code for the successful communication.

SMACS avoids computation and communication overhead of transmitting neighborhood information to a central node by using a local scheme instead of a global assignment. Since the neighborhood discovery process is executed regularly, the protocol assumes to efficiently adapt to all the topology changes. As a densely deployed WSN usually has low traffic load, nodes using SMACS will have highly populated schedules and have to wake up quite often just to discover that there is no packet destined to them. The length of a superframe is also a decisive factor. It should be large enough to accommodate the highest node degree in the network, since a smaller superframe cannot conciliate all the neighbors. With SMACS, the link between two nodes is directional. For bidirectional communication between them, two such links are required.

3.2.3 Channel Polling Based MAC Protocols

With the channel polling scheme, which is also known as preamble sampling and Low Power Listening (LPL), a sending node prefixes data packets with extra bytes called a *preamble*. It sends the preamble over the channel to ensure that the destination node detects the radio activity and wakes up before the actual payload is arrived. On a wake-up, if a radio activity is detected, the receiver turns on its radio to receive data packets. Otherwise, the node goes back to the sleep mode until the next polling interval. To avoid deafness or over-emitting, the sender prefixes preamble at least as long as the *check interval* duration of the receiver to ensure that the receiver wakes up and performs channel sampling at least once while the preamble is being sent [46, 45]. Figure 3.9 illustrates how channel polling works in WSNs.

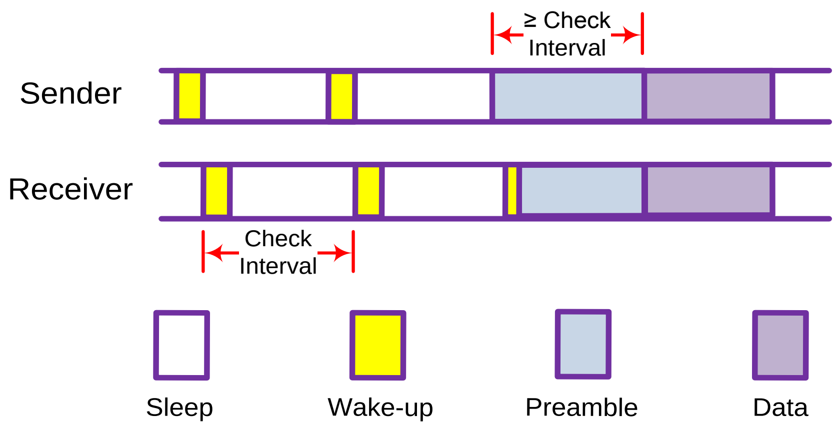


Figure 3.9: Channel polling in WSNs. With the traditional channel polling scheme, the sender first sends an extended preamble, which is at least as long as the check interval of the receiver, to ensure that the receiver will be awake by the time when the data packet is sent.

Since channel polling based protocols do not use common active/sleep schedules, they do not need any synchronization, scheduling, or clustering among nodes. Nodes, specially the receiving ones consume significantly less energy as they wake up for very short period of time to check the availability of the preamble on the channel. Hence, this scheme is considered to be more energy efficient, particularly under low traffic conditions than other schemes [94, 87, 95]. However, the sending nodes pay the price in sending long and extended preambles, and once the receiving node senses the preamble, it stays on and continues to listen until the data packet is received. Moreover, once the non-targeted nodes overhear the preamble, they can distinguish that they are not the addressed nodes only after receiving the complete preamble. This leads to increased transmission and reception lengths and an increased collision probability, which may further increase as the traffic load increases. Another drawback of channel polling is related to the limitation of the duty cycle value. Lowering the duty cycle extends the check interval. That is good from the receiver point of view, but it significantly increases the transmission cost in the shape of long and extended preambles for the sender. Consequently, extended and long preambles cause unnecessary energy consumption both at the receiver and sender ends, overhearing at non-target receivers, excessive latency at each hop, and the bandwidth waste on the broadcast medium. These issues can be tackled by using short preambles, adaptive duty cycle values, and by reducing packet redundancy [96].

Another issue with this scheme is related to the support of radio chip in sending the extended preamble prior to the data packet. Though advanced packet-based radios (e.g., CC2420) frees the MAC layer from handling individual byte unlike byte-level radios (e.g., CC1000), they make channel polling scheme difficult to apply, as they lack the capability of stretching the length of the preamble beyond few bytes [16]. Therefore, for packet-based radios, channel polling is implemented as a burst of contiguous packets, each separated by a short spacing.

3.2.3.1 ALOHA with Preamble Sampling

The combination of ALOHA with the preamble sampling, presented in [46], is considered as the pioneer and typical example of the extended preamble based channel polling scheme. Its basic idea is to let the receiver sleep most of the time when the channel is idle. Like the ALOHA, the sender tries to transmit as soon as it has data to send. However, the sender needs to transmit a preamble of length Tp in front of every packet. The receiver wakes up periodically every Tp seconds and checks for an activity on the channel. If a preamble is found, it stays on and continues to listen the channel until the packet is received. After receiving a successful packet, the node sends an acknowledgment message back to the receiver. The extended preamble ensures that the window of the receiver will be hit during wake-up. Figure 3.10 shows the basic working of ALOHA with preamble sampling technique.

Though the idea behind this technique was to reduce the time spent of ALOHA in listening to an idle medium, it carries all the mentioned problems of the extended preamble technique.

3.2.3.2 B-MAC

The channel polling scheme has been renamed as the LPL in the Berkeley MAC (B-MAC) protocol [45]. Its basic working is same as of the ALOHA with Preamble Sampling technique [46]. However, B-MAC adapts the preamble sampling for the CSMA protocols. A node using B-MAC can have independent *awake* and *sleep* periods. The sum of both these periods is called a *check interval*. While transmitting, the node precedes the data packet with a preamble that is slightly longer than the check interval of the receiver. Nodes wake up at each awake period and sample the medium shortly. If the node detects the preamble, it remains awake to receive the whole preamble. If the preamble is destined to this node, it further

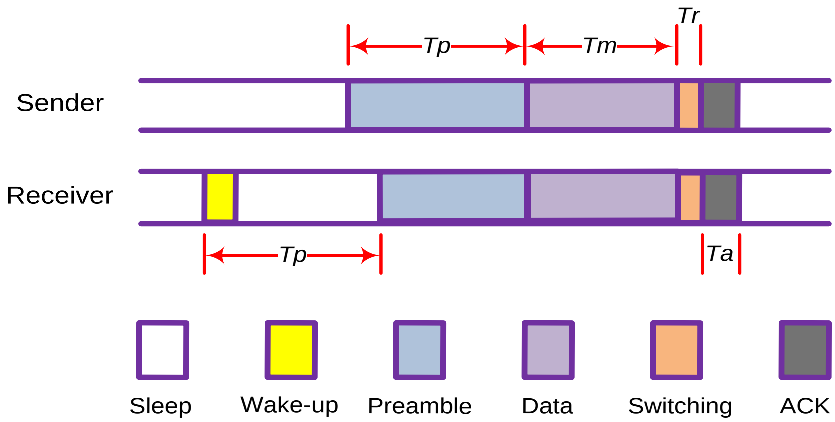


Figure 3.10: ALOHA with Preamble Sampling. The sending node transmits a preamble of length Tp before any data packet. The receiver wakes up periodically every Tp seconds and stays awake if the preamble is found. After sending and receiving a data message for a Tm time, both the sender and receiver switch their radios for a Tr period. An ACK is sent for a Ta time to mark the successful transmission.

extends its wake-up time to receive the data packet; otherwise it goes back to the sleep mode. With the extended preamble, the sender is assured that at some point during the preamble sending, the receiver will wake up and detect the preamble.

B-MAC uses the CCA to determine whether the channel is clear. Assessing the accurate status of the channel is critical for the successful working of the channel polling scheme. Both the *detection failure*, i.e., if the node is unable to detect the preamble that is available on the channel and the *false positive*, i.e, the wrongly detection of the preamble, cause for energy waste and may cause delay and packet loss. Instead of using a threshold, which is the common method for many CSMA protocols, B-MAC improves the quality of CCA by using two phases; *noise floor estimation* and *signal detection*. B-MAC takes signal strength samples during times when the channel is assumed to be idle, such as immediately after a packet transmission. Samples are then stored in a FIFO queue and the median of the queue, M_q, is added to an exponentially weighted moving average with decay α. The noise floor, NF, is then estimated by using the following equation [1].

$$NF = \alpha M_q + (1 - \alpha)\, M_{q-1}$$

Once a fair estimation of the noise floor is achieved, B-MAC uses an *outlier detection* method to search for outliers in the received signal through multiple signal strength measurements. If an outlier is detected, B-MAC declares the channel is clear because a valid packet could never have an outlier significantly below the noise floor. Alternatively, if no outlier is found for five samples, B-MAC declares the channel is busy, and the backoff scheme is then used.

With an extended preamble, B-MAC reduces duty cycle and minimizes idle listening, especially when there are no packet exchanges. B-MAC supports on-the-fly tuning of services by providing bidirectional interfaces to enable or disable them and provides interoperability for the higher layer protocols. In order to adapt check intervals to traffic conditions, B-MAC offers eight low power listening modes corresponding to eight different check intervals. It also draws the optimal listening mode for a node based on the neighborhood size in the network. The efficient CCA helps in eliminating most of the false positives and adapting the node to its surroundings.

However, during transmissions, the preamble sent by the sender needs to be longer than check interval durations of receivers. As a result, B-MAC carries all the stated problems of the extended preamble technique. The modified CCA leads to the increased delay, complexity, and memory usage, since each node needs to have several channel measurements before determining the status of the channel and, it has to keep a statistical track of these measurements.

3.2.3.3 B-MAC+

The B-MAC+ protocol [97], an enhanced version of B-MAC, tries to improve the energy consumption of nodes, particularly of the receiver and overhearing nodes. It divides the long preamble of B-MAC into chunks of *countdown packets*, each one contains a short preamble and a data payload. The data payload contains a *counter* representing the size of the remaining part of the preamble and the *address* of the destination node. A sending node sends out chunks of countdown packets before the data packet. On receiving a countdown packet, the receiver node checks the destination address. If the packet is addressed to the node, it extracts the remaining time from the counter, goes to the sleep mode, and will wake up just before the beginning of the payload transmission time. Alternatively, if the node is not the addressed node, it goes to sleep mode and will continue its normal routine to check the channel.

B-MAC+ significantly reduces the energy consumption at the receiving nodes, but the biggest beneficiaries are the overhearing nodes. An overhearing node goes directly to the sleep mode once it finds that the packet is not destined to it. The receiving node has to go twice through wake-up and sleep cycle during a check interval duration, which limits the energy saving. Like B-MAC, the sending node of B-MAC+ still has to pay the price in sending chunks of countdown packets for at least the check interval duration. Therefore, it does also not improve the packet latency.

Another MAC protocol called the Micro Frame Preamble (MFP)/Data Frame Preamble (DFP) [98] also works on the same approach to B-MAC+ to cut down the preamble length for the receivers.

3.2.3.4 EA-ALPL

The Energy Aware Adaptive Low Power Listening (EA-ALPL) protocol [99] improves the adaptability and non-uniform energy consumption of B-MAC protocol.

As said earlier, B-MAC offers eight different listening modes for a network corresponding to eight different check intervals. However, unlike B-MAC, which sets a network-wide listening mode, EA-ALPL enables each sensor node to set its own listening mode according to its duty cycle and the number of descendants in its routing tree. Thus, EA-ALPL is a cross-layer mechanism, which enables nodes to dynamically learn the listening modes of their neighbors and to choose the appropriate transmit mode in order to reduce idle listening and increase packet reception.

EA-ALPL assumes a proactive routing protocol in which nodes periodically send messages containing updated routing information to their neighbors, which enables a node to adapts its listening mode according to its routing parent. Once nodes learn about their neighbors, a routing graph is formed, which helps a node to know about its descendants. The node sets the optimal listening mode for its current topology position and sends the next route update message. Thus each node in the network always has updated information about their neighbors. An overdriven node can increase its duty cycle to let neighbors to increase its routing cost, which causes its children to choose another parent.

Experiment results that have been achieved using 14 mica2 motes in [99] yield 16% to 55% power saving for EA-ALPL as of B-MAC. However, nodes need to exchange many route messages to keep themselves as well as their neighbors updated. Saving the state information of the neighbors could be a memory and time consuming in a dense WSN. Missing an update route message can be damaging, since it carries important information about neighbors. To overcome this, nodes using EA-ALPL always send their routing updates with the longest preamble, which represents all the problems of the extended preamble technique, particularly if the number of update messages is high. Every time before sending a preamble, a node adjusts its transmit mode in accordance with the routing parent by going through its state information from the routing table, which increases the latency of the packet.

3.2.3.5 WiseMAC

WiseMAC [47] is among the initial protocols working on the non-persistent CSMA combined with the channel polling mechanism. The basic functionality of WiseMAC is more or less similar to B-MAC. However, in order to mitigate idle listening and reduce energy consumption incurred by the long and fixed length preamble, WiseMAC lets a node learn about the awake periods of its neighbors. Nodes learn sampling schedules of their neighbors by piggy backing the remaining time to the next awake period in the ACK messages. They also keep an updated table containing sampling time offsets of their neighbors. A sending node sends the preamble just before the receiving node wakes up, and hence keeps the preamble length at minimum. Clock drifts may make the transmitter to send a long enough preamble to cover up the estimated drift. Figure 3.11 depicts the basic working of WiseMAC.

All nodes in a network sample the medium with the common basic cycle dura-

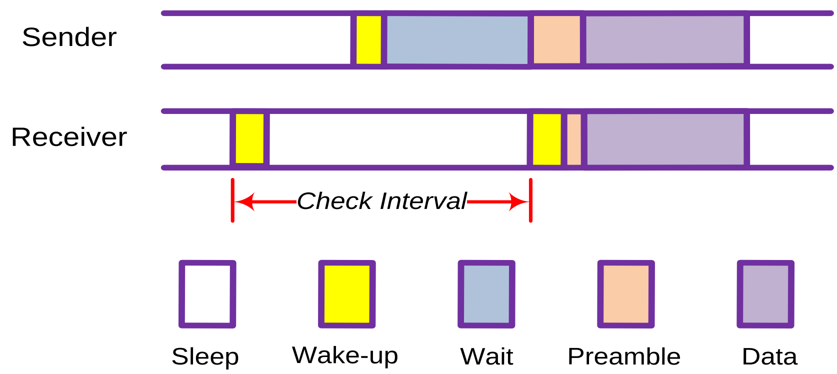


Figure 3.11: The working of WiseMAC. The sending node, knowing the wake-up time of the receiver, waits until the wake-up time of the receiver and then sends a short preamble to intimate it for the forthcoming data packet.

tion. However, their awake and sleep periods are independent and left unsynchronized. WiseMAC uses short preambles for regular traffic and switches to longer preambles for infrequent communication. For very low traffic loads, where data packets can be smaller than the preamble, WiseMAC repeats the data frame instead of the extended preamble.

With WiseMAC, over-emitting can occur if the receiver is not ready at the end of the preamble due to factors such as interference or collision. This over-emitting can increase further with the increase in the preamble and data packet size. As nodes are unsynchronized, keeping wake up times for all neighbors is a memory and time consuming task for a node. In case of a broadcast communication, a transmitter has to deliver the same packet many times to each neighbor. This redundant transmission leads to higher latency and energy consumption for nodes. In addition, the hidden terminal problem can spring up, when one node transmits the preamble to a node that is already receiving packets from another node. WiseMAC does not provide a mechanism to adapt schedules of nodes to varying traffic patterns.

3.2.3.6 RI-MAC

Though the Receiver-Initiated MAC (RI-MAC) [90] is not really a preamble sampling protocol, the reason to put it under this category is that the authors have compared it only with the two preamble sampling protocols; B-MAC [45] and X-MAC [95]. RI-MAC differs from other protocols in how the sender and receiver start communicating with each other. With RI-MAC, the receiver initiates data transmission by sending a short beacon frame to the sender. The sender, after waking up, remains silently active to receive the beacon.

Since no preambles are transmitted on the channel, only the beacon and data packet transmission of RI-MAC can keep the channel utilization at minimum as compared to the preamble sampling protocols. The transmitted beacons can also help in coordinating neighbors, and in result the receiver adaptively varies its channel utilization as per traffic loads. This allows RI-MAC to achieve fair throughput, packet delivery ratio, and power efficiency under varying traffic loads.

Though the idea of shifting initiation of communication from the sender to receiver is interesting, these schemes are in fact similar to the classical one with only the difference of transferring communication burden from one party to another. Therefore, these protocols carry all the discussed problems of overhearing, idle listening, and collision in one or the other way. Moreover, such protocols cannot be used for broadcast or multicast communication.

3.2.4 Hybrid MAC Protocols

Hybrid MAC protocols combine the strengths of two or more different MAC schemes in order to achieve a joint improvement. They usually combine a synchronized scheme with an asynchronous one. Though hybrid protocols aggregate the advantages of multiple schemes, they also carry, among other, scaling and complexity problems in maintaining two or more different working modes.

3.2.4.1 IEEE 802.15.4

The IEEE 802.15.4 Task Group 4 (TG4) [42], together with the ZigBee Alliance [100], has developed a protocol stack for a LR-WPAN . IEEE 802.15.4 provides the functionalities of the PHY and MAC layers for the LR-WPAN. Although 802.15.4 was not exclusively designed for WSNs, its several features such as simplicity, low rate, low power, and low cost communication has made it a strong candidate for the WSN communication stack. IEEE 802.15.4 MAC protocol combines the contention based channel accessing scheme with the scheduling based one. The detailed working of 802.15.4 protocol and its layers is given in Chapter 5, where we also discuss the limitations and possible enhancements to the protocol for energy, latency, and bandwidth critical WSN applications.

3.2.4.2 Z-MAC

The Zebra MAC (Z-MAC) [86] protocol is a hybrid scheme that combines the strengths of TDMA and CSMA while offsetting their weaknesses. Z-MAC is a traffic adaptive protocol in the sense that under low contention, it switches to CSMA to achieve high channel utilization and low delays, and under high contention, it switches to TDMA to achieve high channel utilization, fairness, and less collisions. Unlike the traditional TDMA scheme, a node using Z-MAC can also utilize slots assigned to other nodes. However, the owner of the slot always has higher priority over the others, which reduces the chances of collisions. The owner selects a random backoff time of up to Ta, whereas other nodes select a backoff time between Ta and Tb, where $Ta > Tb$, for contention within the time slots of other nodes.

To reduce the hidden terminal effect during high contention, a node using Z-MAC sends an Explicit Congestion Notification (ECN) frame to the neighbor it has message for. Note that nodes observe the contention level by tracking the time they spend in backoff due to the unsuccessful carrier sensing. The neighbor further broadcasts the ECN to its neighbors. On reception of the ECN message, a node enters a High Contention Level (HCL) state, where it only attempts to transmit in its slot and those of its direct neighbors, thus the node helps in reducing contention between neighbors two hops apart. The node switches back to the Low Contention Level (LCL) state if it does not receive an ECN within a time period.

Z-MAC uses CSMA as the baseline MAC scheme, and in the worst case it always falls back to CSMA. Z-MAC provides a simple two-hop synchronization scheme, where each sending node adjusts its frequency based on its current data rate and resources. Hence, Z-MAC is robust to timing and slot assignment failures, channel conditions, and topology changes. However, during the start up phase, Z-MAC requires global time-synchronization, which certainly is an energy, time, and memory consuming for lightweight nodes. A highly dynamic WSN may need to perform this costly procedure more than once. Apart that, complexity in maintaining both CSMA and TDMA modes, switching between the HCL and LCL states, contention and possible collisions among nodes to gain access of the slots owned by other nodes, bandwidth under-utilization, and distribution of ECN messages under high local traffic (due to the occurrence of an event) are also the issues with Z-MAC.

3.2.4.3 SCP-MAC

The Scheduled Channel Polling MAC (SCP-MAC) [87] is another hybrid protocol that combines channel sampling with TDMA in order to minimize the preamble length. Nodes using SCP-MAC perform channel polling periodically and switch to sleep mode when there is no traffic available on the channel. However, unlike the channel polling scheme, sampling times of nodes are synchronized. In result, a very short *wake-up tone* can be sent to wake up the receiver, which largely reduces the overhead of transmitting long preambles.

The basic working of SCP-MAC is illustrated in Figure 3.12. Before communicating, the node silently waits in sleep state and performs carrier sense within the first contention window (CW1) just before the polling time of the receiver. On finding the channel idle, the sender sends a short wake-up tone to activate the receiver. Otherwise, it goes back to sleep and performs regular channel polling. After the sender wakes up a receiver, it enters the second contention window (CW2) and sends data if the node still detects the channel idle. The receiving node that has received a wake-up tone extends its wake-up period to receive the data packet.

SCP-MAC uses two separate contention phases to achieve low collision probability. The RTS/CTS handshake can be enabled or disabled with SCP-MAC. When the RTS/CTS is disabled, overhearing is performed by examining destination address from the packet header. Adaptive listening is also supported in SCP-MAC,

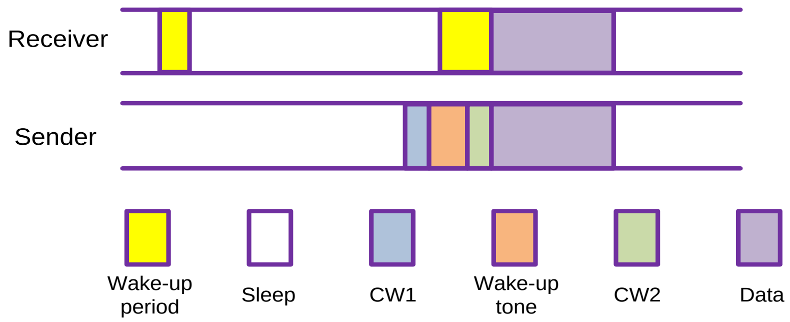


Figure 3.12: Basic working of SCP-MAC. A sender waits in sleep mode until the wake-up time of the receiver and then sends a wake-up tone to wake the receiver up before sending a data packet. To minimize the collision probability, it performs carrier sense twice; one before sending a wake-up tone and the other before sending the data packet.

where the MAC layer, after transmitting a packet, immediately polls the channel for additional traffic. In order to avoid schedule based delays with SCP-MAC, a node can coordinate the schedules of all nodes along a path by using the fast-path schedule allocation. Results shown in [87] interpret that SCP-MAC can extend network lifetime of a WSN by a factor of 2 to 2.5, can reduce the duty cycle by a factor of 10, and can reduce delay by avoiding extended preambles.

However, along with the synchronization overhead and complexity in maintaining both channel sampling and TDMA modes, SCP-MAC also has other drawbacks. All nodes in the neighborhood wake up simultaneously, therefore unintended receivers cannot avoid overhearing packets (particularly preambles) and participating in higher contention at that time. Collisions may occur, which force nodes to postpone their transmission to the next synchronized wake-up time and thus inducing higher latency and energy consumption for nodes. Double contention also increases the system latency.

3.2.4.4 Funneling MAC

In WSNs, packets generated by nodes usually travel hop-by-hop in a many-to-one pattern, i.e, from nodes to the sink node. Thus, they exhibit a unique funneling effect, which could lead to an intensified traffic, collisions, delays, and energy drain as events move closer toward the sink node. A hybrid, localized, and sink-oriented Funneling-MAC protocol [88] exposes this WSN phenomenon. Funneling-MAC is mainly a CSMA/CA protocol with a localized TDMA algorithm, which works only within a few hops from the sink node in the *intensity region*.

The basic working of Funneling-MAC is shown in Figure 3.13. The TDMA scheduling is managed by the sink node and operates locally in the intensity region. The burden of computing and maintaining the depth of the intensity region also falls on the sink node. The sink node broadcasts a beacon that triggers the TDMA scheduling. All nodes perform CSMA by default unless they receive the beacon and are then considered to be in the intensity region and called *F-nodes*. The area

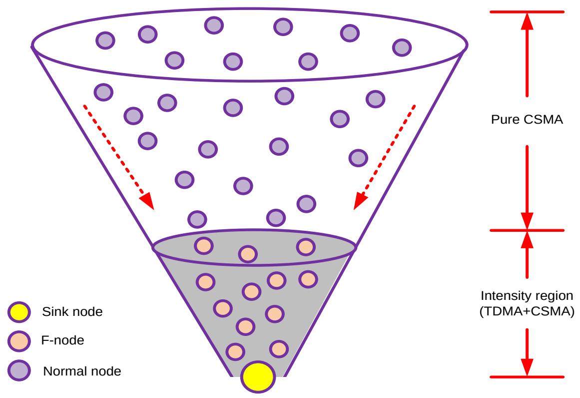


Figure 3.13: Basic working of Funneling-MAC. A nodes that receives a beacon inside the intensity region is called the F-node. The area of the intensity region is defined by the sink node by controlling the transmission power of the beacon.

of the intensity region is defined by the sink node by controlling the transmission power of the beacon. The F-nodes switch back to CSMA, if they do not receive the beacon within the specified time period. The sink node calculates the TDMA schedule as per traffic conditions. Each F-node transmits its scheduled packet at the allocated time slot specified in the TDMA frame. In order to allow the transmission of data packets that have not been allocated slots yet, a CSMA frame is reserved between two consecutive TDMA frame schedules.

With Funneling MAC, F-nodes consume more energy in receiving several beacons and schedule frames. Beacon and schedule frames are sent at potentially high power and could interfere with on-going communication. Many duty cycled nodes within the intensity region could become interferes by missing out the beacon if their periodic wake-up times do not match with the beacon time. Funneling MAC also carries complexity in managing intensity region over the time and in performing CSMA after every second TDMA frame. The number of F-nodes in a dense WSN could be very high, and in result each F-node has to wait longer for its turn to communicate with the sink. This will increase network latency.

3.3 Comparison of Different WSN MAC Protocols

Having discussed the detailed working, advantages, and drawbacks of the major channel accessing categories and MAC protocols falling in each category, we next present a comprehensive comparison of these protocols in Table 3.1. Most of the time, it is hard to compare protocol architectures that use different working space, we in the table, however, analytically compare all the discussed MAC protocols for their support to some metrics that are vital in designing any WSN MAC protocol. The considered metrics are related to the energy efficiency, timeliness, asynchrony, adaptivity, and scalability factors. Chapter 4 further details these metrics.

Table 3.1: Comparison of different MAC protocols proposed for WSNs based on their support to various parameters

Protocol	**Type**	**Energy Eff**	**Delay Eff**	**Async**	**Adaptive**	**Scalable**
S-MAC	CSMA	yes[1]	no	yes[2]	yes[1]	yes[1]
T-MAC	CSMA	yes[1]	no	yes[2]	yes	yes[1]
DSMAC	CSMA	yes[3]	yes[1]	no	yes	no
GSA	CSMA	yes[1]	yes[1]	no	no	no
STEM	CSMA	yes[4]	no	yes	no	yes
LEACH	TDMA	yes	no	no	no	no
TRAMA	TDMA	yes[1]	no	no	yes	no
SMACS	TDMA	yes[1]	no	yes	yes[1]	yes
ALOHA-PS	CP	yes[1]	no	yes	no	yes
B-MAC	CP	yes[1]	no	yes	yes[1]	yes
B-MAC+	CP	yes[1]	no	yes	no	yes
EA-ALPL	CP	yes[7]	no	yes[7]	yes[7]	yes[7]
WiseMAC	CP	yes[1]	yes[1]	yes	yes	yes
RI-MAC	CP[6]	yes[1]	no	yes	yes	yes
802.15.4	Hybrid	yes[1]	yes[1]	yes[5]	no	no
Z-MAC	Hybrid	yes	no	no	yes	no
SCP-MAC	Hybrid	yes[1]	yes[1]	no	yes	no
Funn-MAC	Hybrid	yes[1]	yes[1]	yes[6]	yes	yes

CP = Channel Polling
[1] Suboptimal, can be improved
[2] Uses common listen periods
[3] Trades-off with other parameter
[4] Uses two different channels
[5] Depends on the topology
[6] Supports partially
[7] Uses frequent update messages

It is clear from the table that most of the protocols do not support all of these parameters. They either support a parameter partially or trade-off with another parameter, particularly in favor of energy. Several protocols are suboptimal for a given parameter and can be improved further. Significantly, most protocols do not consider delay as an important metric in their design.

3.4 Conclusion

Why and how WSN MAC protocols are different from those of other distributed wireless systems has been articulated in Chapter 2. Throughout this chapter, we have dwelt on different channel accessing schemes used by these MAC protocols and categorized them into four classes; contention based, scheduling based, channel polling based, and hybrid protocols. Along with the detailed working of each category and protocols falling in, their advantages as well as disadvantages have also been discussed in detail. It is obvious that although a plenty of WSN MAC protocols have been proposed over the last years, the selection of a proper MAC protocol mostly depends on the application.

Contention based MAC protocols usually follow common active/sleep schedules, where all neighboring nodes wake up simultaneously at fixed periodic intervals to access the channel through CSMA. These protocols are scalable, robust to network dynamics, and do not need any clustering or topology information. However, traffic rigidity, latency, collisions, overhearing, idle listening, and hidden and exposed terminals are the issues with these protocols. Many of these issues have been avoided in scheduling based protocols, where nodes are forced only to access their allocated slots. But, overhead and extra traffic needed to setup and maintain schedules and synchronization, higher queuing delays, adaptivity to traffic and topology changes, and scalability are the major concerns with scheduling based schemes.

The channel polling based scheme is getting more attention from researchers as they concede substantial energy savings than the other schemes. Nodes under this scheme neither follow common active/sleep schedules nor they need any sort of synchronization or clustering. However, this scheme still suffers from several limitations causing from the use of the long preamble. A preamble sent by the sender needs to be longer than the check interval of the potential receiver to guarantee that the receiver will be awake when the data portion arrives. These extended preambles concede several energy and delay related problems not only for the sender and receiver but also for the non-targeted/overhearing receivers. We concluded that although the energy consumption of the channel polling scheme is lower than other schemes, the implications of extended preambles must be alleviated. This can be achieved by using short preambles and adaptive duty cycle values, and with reduced redundancy. Our proposed MAC protocol in Chapter 6 incorporates all these factors in order to provide an improved energy and delay performance for a WSN.

After discussing several protocols from each category, we have presented a substantial comparison of these protocols for different metrics in Table 3.1, which shows that most of the protocols lean towards the energy efficiency factor. They either support other parameters partially or trade them off with another parameter, particularly in favor of energy. Some of the protocols are suboptimal for a given parameter and can be improved further. Importantly, many of them do not consider latency as an important metric in their design. With the advent of new MEMS and energy harvesting techniques, one may expect a vital breakthrough in energy domains in the near future. Therefore, we believe that other performance criterion, especially latency based QoS, may play critical role in the future applications of WSNs. Our proposed AREA-MAC protocol considers and also provides significant results not only concerning the energy efficiency but also the latency, asynchrony, packet delivery, scalability, and adaptability to traffic and topology changes in a WSN.

Chapter 4

Evaluation Methods and Metrics

The desire to know is far more important than achievement and/or performance measures.

–Caine & Caine

THIS chapter gives an overview of the evaluation methods and metrics that have been used throughout this thesis. The simulation and real experimentation platforms are also briefly discussed. As stated earlier, the contribution of this thesis is to design, implement, and evaluate a MAC protocol, which targets at the reduction in energy consumption and packet latency for sensor nodes.

4.1 Evaluation Methods

Merit or technical capabilities of any system/protocol are validated through evaluation. Selecting an appropriate evaluation method and correct performance measures is core to achieve the desired output of a system. Performance evaluation not only helps in determining capabilities of the system against state of the art techniques but also directs for any possible enhancement. Efficient evaluation helps in isolating effects of different parameters and interpreting results, so that meaningful statements about the system can be made. Sometimes the system evaluation, however, also depends on several (non-technical) factors such as time, budget, background of the research team, and available facilities. Three important and often necessary methods to evaluate and validate any system are the *analytical modeling*, *simulation*, and *real experimentation* methods [40, 41]. In this work, all three methods have been used to evaluate the proposed scheme. Additionally, the optimization evaluation method has also been used to achieve an optimal duty cycle value for nodes that provides minimum energy consumption and minimum packet delays for a WSN. The below is the brief description of the used techniques.

4.1.1 Analytical Method

The mathematical analysis can provide a preliminary response to a system, which can be followed and further strengthen by the simulation and/or real experimentation methods. The analytical method generally provides a good insight into the effects of various parameters, their ranges, and interactions. It is the cheapest evaluating method as no hardware/software are usually required.

In this work, we first propose several improvements related to energy, delay, and bandwidth factors to IEEE 802.15.4 MAC protocol in Chapter 5. We apply an analytical method to formulate and validate our improvements before presenting them in a graphical form. Subsequently in Chapter 6, we mathematically formulate several models for energy, delay, network, and traffic behavior of sensor nodes for AREA-MAC protocol. We also analytically compare AREA-MAC with the long preamble sampling method and prove that AREA-MAC provides improved performance, especially in terms of energy and delay for a WSN.

4.1.1.1 Optimization Method

Mathematical optimization models, which are widely used in production planning, engineering design, financial portfolio selection, and many other decision-making areas, achieve the best (either maximum or minimum) outcome for an *objective function* in a given *set of constraints*. A constraint set is expressed in the form of equations or inequalities. Such models are expressed in the following canonical form [101, 102]:

$$\begin{aligned} &\max \quad c^T x \\ &\text{subject to:} \\ &\qquad Ax \geq b \\ &\qquad x \geq 0 \end{aligned}$$

Where x is a vector of variables that are to be determined, c and b are vectors of known coefficients and A is a known matrix of coefficients. The equations $Ax \geq b$ are the constraint over which the objective function ($c^T x$) is to be optimized. Any vector x satisfying the set of constraints is called a *feasible solution* of the problem. The *optimal solution* of the problem yields a unique maximum (or minimum) value of the objective function. If a feasible solution of the problem does not exist, the problem is assumed *infeasible*.

In order to achieve optimal solution in terms of energy and delay and to find out a trade-off between them, we present three different optimization problems in Chapter 8. These optimization problems correspond to minimization of energy consumption, minimization of end-to-end delay, and maximization of overall system lifetime of an AREA-MAC network. We solve the first two problems for an optimal duty cycle value for a sensor node and then validate our results by comparing them with the simulation results.

4.1.2 Simulation Method

Simulation is a fast, easy, cheap, and often very useful technique to analyze a system, particularly in the ever-increasing and complex field of communication networks. Importantly, simulation can help in predicting the performance of the system during the design stage. Most of the computer based systems are analyzed via simulation that provides the system many other alternatives to choose and compare with. Moreover, the system can easily be analyzed under a wide variety of workloads and environments via simulation that otherwise might be left undone. However, a careful insight is needed to select values of different parameters and assumptions for a simulation system as simulation can produce premature or misleading results [40]. Moreover, a system represented with a simulation is only an *abstract model*, which usually provides selected and specific features under an ideal environment/scenario [41].

In Chapter 7, we implement AREA-MAC on *OMNeT++ simulation* platform [48] by using *Mobility Framework* [103]. A brief introduction of OMNeT++ and Mobility Framework is given below.

4.1.2.1 OMNeT++

The Objective Modular Network Testbed in C++ (OMNeT++) [48] is an object-oriented, extensible, and component-based discrete event network simulator, which is widely used to simulate several communication, queuing, and other distributed networks. In discrete event based networks, the system state is changed only at discrete points in time with the occurrence of an event. The component-based architecture of OMNeT++ facilitates designer to map devices, protocols, or channels into model components (modules), which can be combined together or detach from each other at any stage in the same way as a child acts with LEGO blocks. A comparison presented in [104] shows that OMNeT++ outperforms another widely used ns-2 [105] simulator in almost every aspect. OMNeT++ provides, among others, the following features:

- C++ kernel and an extensive simulation library to schedule events, send and receive messages, and configure and assemble modules
- Rich graphical interface for designing, running, evaluating, debugging, and tracing simulation runs
- Support for hierarchical nested modules with an unlimited depth
  - The top level module is called a *system module*, which contains a lowest level *simple module* or a *compound module* containing more than one simple modules, as depicted in Figure 4.1
  - Modules have *gates* as input and output interfaces
  - Modules communicate by exchanging *messages*

  - Behavior of simple modules is programmed in C++

- NEtwork Description (NED) language for topology description
- Support for extensive random number generating distributions
- Support for parallel simulation
- Collection, analyzation, and visualization of several types of results
- Availability of many simulation models such as INET Framework, INET-MANET, Mobility Framework (recently integrated with MiXiM), Castalia, OverSim, and many more

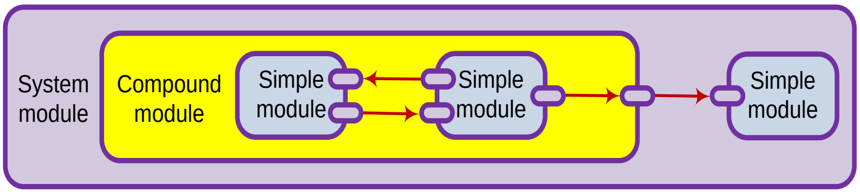


Figure 4.1: OMNeT++ Module Hierarchy. Simple modules are active elements in an OMNeT++ model and cannot be derived further.

4.1.2.2 Mobility Framework for OMNeT++

Mobility Framework (MF) [103] provides several OMNeT++ simulation models for fixed and mobile wireless distributed networks. It consists of an architecture supporting a dynamic connection management, which fits well with duty cycled sensor nodes. The main design goals of MF are easy *usability* with simple and well defined interfaces, *extensibility* with efficient module integration, and *scalability* to support many nodes with reasonable execution time and memory usage [106]. In order to provide the stated functionalities, MF contains many OMNeT++ modules and interfaces. We below present some of them that are relevant to the implementation of our MAC protocol.

The *ChannelControl* module of MF depending on the distance and physical characteristics of nodes dynamically sets up, maintains, and tears down all potential connections between nodes. Thus it enables a node to receive every data packet that its transceiver is potentially able to sense. The physical layer then decides dependent on the received signal strength whether it is a valid packet or it should be treated as noise. Upon initialization, the *ChannelControl* module determines the maximum interference distance based on parameters such as the carrier frequency of the channel, transmission power of nodes, and other propagation specific parameters, and then calculates the connections between all nodes and updates them every time a change in the network occurs.

Each layer of the communication protocol stack, shown in Figure 2.4, has its own modules, which perform the corresponding functionality. At the application layer, an application can generate either one or a burst of broadcast messages, which are then forwarded to the lower layers. Every layer adds the header information of the *headerLength* size to the packet and forwards it to the next layer. The respective layer at the other end extracts the relative header information and then removes the header from the packet. A *Network Interface Card (NIC)* compound module that is depicted in Figure 4.2 provides the functionalities of both PHY and MAC layers. At the PHY layer, it has two submodules, namely a *snrEval* and a *decider*, whereas the MAC layer module is called a *macLayer*.

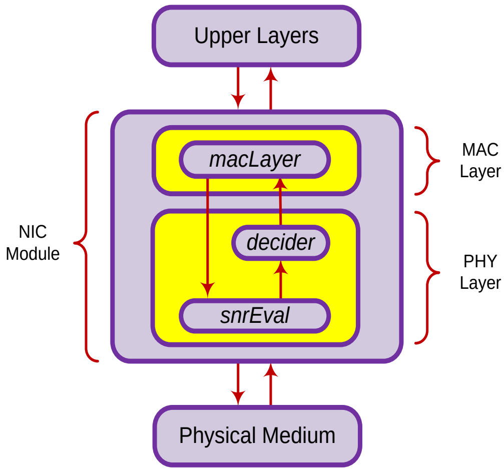


Figure 4.2: The NIC Module of the mobility framework. Due to the strong coupling of the MAC and PHY layers, the NIC module combines the functionalities of both layers.

The *snrEval* module simulates the transmission delay for all received messages and also calculates SNIR information. The *decider* module can use this information to decide whether a message is lost, has bit errors, or is correctly received. However, messages received from the upper layers are directly bypassed to the *snrEval*. The parameters needed to compute SNIR and attenuation of a signal are *transmitterPower*, *carrierFrequency* and *pathLossAlpha*. Note that the *ChannelControl* module also has versions of these parameters (*pMax*, *carrierFrequency*, *alpha*) but they are only used to compute the interfering distance of nodes. A user can define the Signal Attenuation Threshold (SAT) value for the required signal strength received power levels. A signal weaker than SAT is simply neglected. The *macLayer* module works in close coordination with both *snrEval* and *decider* and enables a node to sense the carrier and to reliably communicate with other nodes.

In addition, MF also has a *Blackboard* module that provides a cross-layering functionality by offering an information exchange among different layers. This information may include the status of the radio, current energy level, or the display appearance of a node. For example, a MAC protocol while sensing the carrier inquires from the PHY layer via the blackboard about the status of the carrier or it distinguishes with the help of *Blackboard* module whether the radio is in sleep, receive, transmit, and/or idle mode.

Though the basic MF only supports deterministic radio propagation based Free Space and Two Ray Ground models, the extension provided in [107] enables MF to support several probabilistic propagation models such as Long Normal Shadowing, Nakagami, Rayleigh, and Rice models. The description of each of the mentioned model is given in [107–109]. Most of the wireless simulation models consider deterministic propagation (unit disk graph) models and determine signal strength of each arriving frame usually based on the distance parameter only. These are unrealistic models that usually consider, among others, the following assumptions [108]:

- The transmission range of a radio is circular
- All radios have same range
- All radios behave symmetrically (If I can hear you, you can hear me)
- All radios behave optimistically (If I can hear you at all, I can hear you perfectly)

Probabilistic models provide more realistic modeling of radio propagation by calculating the individual reception power of each arriving frame with the help of a probability distribution with the average reception power as one of the parameters [107]. For the simulation of AREA-MAC, we use a probabilistic based Nakagami propagation model in Chapter 7, which also further details about the simulation environment, simulation models, and selection of parameters for the implementation of AREA-MAC protocol.

4.1.3 Real Experimentation Method

After analytical and/or simulation evaluations, a system can finally be tested and validated via the real experimentation method, which usually authorizes the pre-deployed results obtained via either simulation or analytical (or both) methods. Real experimentation results are mostly preferred over other methods, but more efforts are commonly required to achieve them. This evaluating method needs more time, hard work, and budget in purchasing, installing, and maintaining required hardware, tools, and staff. Real experimentation can provide improved/changed results as of other methods and sometime it becomes difficult to diagnose whether this is because of a change in environment or any specific parameter setting.

In Chapter 9, we implement AREA-MAC on the DES-Testbed [49] that uses MSB-A2 sensor nodes [10]. A brief description of their hardware characteristics is given below, while the testbed topology and parameter selection is detailed in Chapter 9.

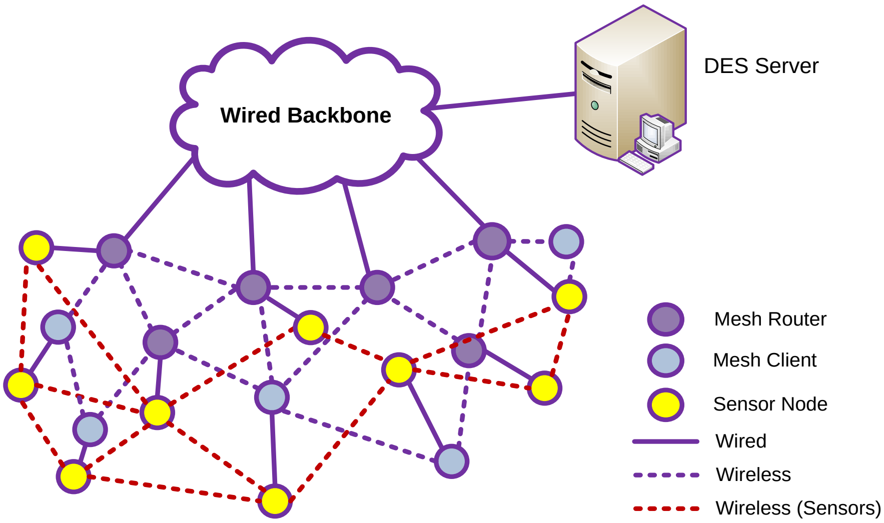


Figure 4.3: Architecture of the DES-Testbed [49]. The stationary mesh routers build the core backbone network. A mesh client, which can be a mobile, is connected to the mesh backbone. Each sensor node is connected by wire to a mesh router or a mesh client but communicates with other sensor nodes over a wireless channel. The DES server provides NFS, scheduling, database, and other management services to DES-Nodes.

4.1.3.1 DES Testbed

The Distributed Embedded Systems Testbed (DES-Testbed) [49] is a hybrid wireless testbed deployed across the multiple building campuses of Freie Universität Berlin. It consists of a WMN and a WSN, both are assembled into an enclosure that is called a *DES-Node*. Currently, the testbed contains more than 100 wireless mesh routers equipped with three or more IEEE 802.11a/b/g network adapters and an equal number of MSB-A2 sensor nodes. Several network architectures and protocols are being experimented and evaluated on the DES-Testbed by using a number of frameworks developed to support the holistic experimentation processes. Figure 4.3 shows the architecture of the DES-Testbed. The mesh routers are based on the PC Engines Alix2c2 embedded board with a size of 152.4x152.4 mm. Table 4.1 presents the hardware details of a mesh router.

Table 4.1: Hardware details of the Mesh router used in the DES-Testbed [49]

CPU	500 MHz AMD Geode LX800
DRAM	256 MB DDR DRAM
Ethernet	2 Ports (Via VT6105M)
Expansion	2 Mini PCI slots & dual USB ports
Storage	CompactFlash socket
Enclosure	Customized TEKO AUS23

4.1.3.2 MSB-A2 Sensor Node

Each of the mesh router of the DES-Testbed is equipped with a MSB-A2 sensor node [10], which is connected via a USB connector. USB connectivity provides power supply to the node and is also used to write firmware images to the flash memory. MSB-A2 nodes operate on a different frequency band than that of WMN and thus set up a parallel testbed. Main components of a MSB-A2 node are highlighted in Figure 4.4, whereas Table 4.2 gives an overview of its hardware architecture. The frequency of the 32-bit ARM7 TDMI-S core based microcontroller can be varied as per application and energy requirements. By using either simple GPIO pins or an on-board mini USB port, extensions can be connected to the MSB-A2.

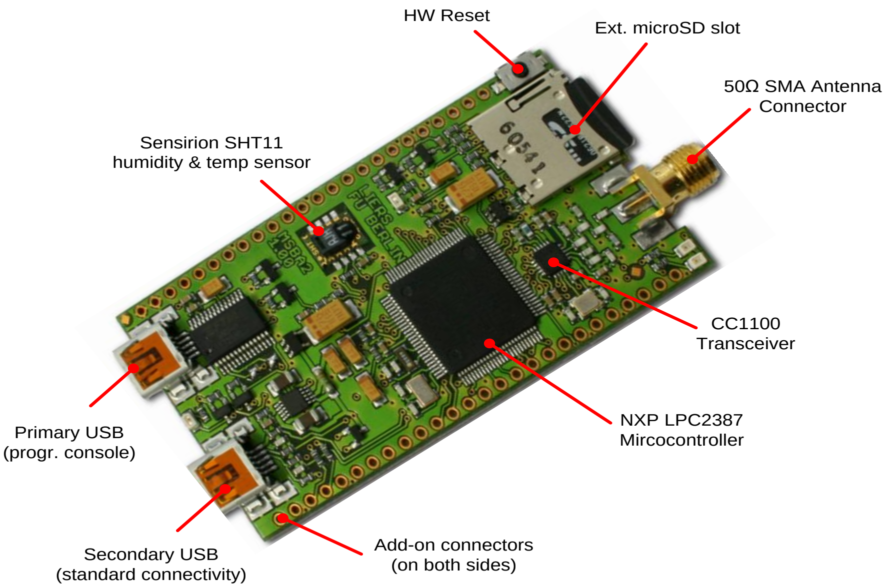


Figure 4.4: Hardware details of the MSB-A2 sensor node. Along with the internal 512 KiB flash memory, external microSD cards can be used. The SHT11 sensor provides humidity and temperature measurements with decent accuracy and low power consumption.

Table 4.2: Hardware Characteristics of the MSB-A2 sensor node used for the DES-Testbed [49]

Microcontroller	NXP Semiconductors LPC2387
CPU Frequency	up to 72 MHz
RAM	98 KiB
Flash	512 KiB
Transceiver	Chipcon CC1100
On-board Sensors	Humidity & Temperature
Expansion	GPIO pins, mini USB port
Storage	microSD card socket

The MSB-A2 node is equipped with a Chipcon CC1100 transceiver [110], which is a small-size, low-power, and low-cost transceiver making it perfectly suitable for several WSN applications. Following are some of the main features of CC1100 [110]:

- Frequency bands of 300-348 MHz, 400-464 MHz, and 800-928 MHz, which includes the ISM and SRD bands
- Integration with a highly programmable modem that supports 2-FSK, GFSK, MSK, OOK, and ASK modulation schemes and data rate from 1.2 to 500 kBaud
- Extensive hardware support for packet handling, data buffering, burst transmissions, CCA, LQI, and WOR modes
- 64-byte transmit/receive FIFOs
- High sensitivity of -111 dBm at 1.2 kBaud, 868 MHz, 1% packet error rate
- Low current consumption: 400 nA in Sleep and 14.4 mA in RX at 1.2 kBaud, 868 MHz
- Fast startup time: 240 μs from sleep to RX or TX mode
- On-chip support for sync word detection, address check, flexible packet length, and automatic CRC handling
- Digital RSSI output
- Programmable channel filter bandwidth, carrier sense indicator, and PQI for improved sync word detection

4.2 Performance Metrics

To quantify and qualify services that are offered by any system or protocol, a set of performance criteria or metrics must be selected. We select the following metrics to evaluate AREA-MAC protocol.

4.2.1 WSN Lifetime

By far the undoubtedly biggest issue in WSNs is energy conservation. As said earlier, sensor nodes are battery operated and recharging or replacing them is often infeasible. This makes energy a very scarce resource that must be wisely managed in order to extend the WSN lifetime. It is difficult to have a precise criterion to define the WSN lifetime. Normally the lifetime of any network is assumed as the time duration from its deployment to the moment when the network turns non-functional [111]. However, when a WSN should be considered non-functional is totally application specific. Hence, many definitions of the WSN lifetime exist in the literature, some of them are concisely sketched below [112, 113]:

- the time the first node dies
- the time a certain fraction of nodes dies
- the time the first cluster head dies
- the time all nodes in a network die
- the time the network breaks in two or more segments
- the time network connectivity and coverage are doomed
- the time each of the target is not covered by at least one or K nodes
- the time the network is not able to provide application specific tasks
- the time the packet reception drops below a threshold
- the time no communication backbone exists

Despite of having many explanations for WSN lifetime, it basically and strongly depends on the lifetimes of the individual nodes that form a WSN. In order to increase the WSN lifetime, the energy consumption of each node must be reduced. Therefore, we use a more general approach of the WSN lifetime and calculate the time the radio of a node spends in different modes for AREA-MAC protocol. As explained in Section 2.4.2, a node consumes substantially higher energy in communication than in sensing or processing. Therefore, we analyze energy consumption of a node by calculating the *time in sleep mode*, *time in receive mode*, *time in transmit mode*, and *time in Carrier Sense (CS) mode* for its radio. Once these times are obtained and with the given initial energy level and an estimated consumption rate, remaining energy of a node can easily be calculated. For this purpose, the *profiling* technique, which is discussed in [114], can also be used.

4.2.2 End-to-End Delay

Since one of the objectives of AREA-MAC is to reduce packet latency for sensor nodes, it offers several concepts such as *pre-ACK*, *adaptive duty cycling*, *message priority*, *short and long sleep modes*, and *message routing* for nodes that significantly reduce packet delay. All these concepts are detailed in Chapter 6. We calculate end-to-end delay for each data packet that a node generates. This is equal to the time a data packet is generated at the application layer and is put in the queue to the time it successfully arrives at the sink node. Every intermediate (forwarding) node first keeps the packet in the buffer and then on finding the channel clear forwards it further.

4.2.3 Packet Delivery Ratio

A node using AREA-MAC generates a data packet randomly (with a given data generating interval and deviation) at the application layer, which is processed down to other layers before being broadcasted at the physical medium. Once the packet has been transmitted by a sender, nodes keep forwarding it until the sink node receives it. We calculate *total data packets generated*, *total data packets sent*, and *total data packets received* by each node. Based on these metrics, we calculate the *packet delivery ratio* of the network as the ratio of total packets generated by all nodes to the total packets received at the sink node.

4.2.4 Number of Preambles Sent and Received

Unlike the long preamble sampling technique, where a node always sends a long and fixed size preamble, a node using AREA-MAC sends short preambles with a short spacing in between. It stops sending further preambles as soon as it receives the pre-ACK from the intended receiver. This reduces the preamble sending and receiving durations for nodes, which ultimately increases the network lifetime and reduces the WSN latency. To confirm this improvement, we calculate the *number of preambles sent* and *number of preambles received* by each node for the whole experiment duration. Note that during the initialization stage, a node based on its duty cycle value calculates the *burst-count*, i.e., the maximum number of preambles to be sent within a check interval duration. Whenever it has a data packet in its buffer to be sent/forwarded, it keeps on sending preambles unless either it receives the pre-ACK or its preamble counter reaches the burst-count. In the later case, the node performs duty cycle, goes to a short sleep mode, and continues the same process at the next check interval.

4.2.5 Number of Overheard Packets

In order to analyze how much energy and time a node spends in processing unnecessary packets, we calculate the *number of overheard packets* for each node. An overheard packet can be a preamble, data, or an ACK packet depending on the radio state in which the node receives it. After sending a preamble, the node enters the *WAIT-FOR-ACK* state, where it only accepts the ACK frame. Similarly, after sending the ACK frame, the node enters the *WAIT-FOR-DATA* state to receive the data frame. A node cannot receive a data frame directly without receiving a preamble and transmitting an ACK frame. However, the sink node can receive a data packet in all states. A node counts all the frames as overheard for that it is not waiting. On a broadcast wireless medium, the higher number of overheard packets is one of the critical perpetrator that drains node energy. Next to the short preamble mechanism, AREA-MAC provides a simple scheme that is explained in Chapter 6 to minimize the redundancy of data packets that further quashes the number of overheard packets for nodes.

4.2.6 Scalability

Since a WSN may contain a varying number of nodes at different times, scalability is a factor that guarantees that its performance does not substantially degrade as the network size varies. For our simulation experimentation, we first consider a WSN having 16 nodes and then increase this number to 36 to analyze how the network adjusts itself with the increasing number of nodes. We then compare and analyze results for several metrics for both network sizes.

4.2.7 Adaptability with Varying Traffic and Topology Conditions

WSN is a dynamic arrangement of nodes where the network topology and traffic may vary frequently due to duty cycling, node failure, channel fading, energy depletion, or random event occurrence. To observe the adaptability to accommodate topology and traffic changes, AREA-MAC is evaluated under different traffic and topology scenarios. Along with the adaptive duty cycle option, where nodes adjust their duty cycle values as per traffic load and message priority of their neighbors, AREA-MAC considers two types of network topologies, i.e, *grid topology* and *random topology* and two types of traffic generating scenarios, i.e., *simple traffic* and *burst traffic*. Normally sensor nodes are deployed randomly in a sensing field, however the grid topology is also practicable, particularly in the pre-deployment of WSNs at research, medical, or other places. The topology and traffic models are detailed in Chapter 6.

4.2.8 Duty Cycle Effect

The selection of the proper duty cycle value for the application is of the essence to the WSN performance. Long sleep periods can save more energy at the cost of higher latency and lower throughput. Short sleep periods result in frequent switching of the radio between on and off modes, which could also outweigh the advantages of duty cycling. For the AREA-MAC evaluation, we use varying duty cycle values for nodes and then compare results for several metrics for each duty cycle value. We then achieve an optimal duty cycle value through optimization that offers maximum energy saving, minimum packet delay, and an acceptable packet delivery ratio for a WSN.

4.2.9 Cross-Layering

Though AREA-MAC is mainly associated with the MAC layer, it also exploits simple routing at the network layer and interacts with the application as well as PHY layer. Thus, AREA-MAC adapts a simple cross-layering mechanism and examines the interaction between different layers. At the application layer, it uses two type of traffic generating scenarios, namely simple and burst traffic. With the simple traffic scenario, a node generates a random number of data packets with the given data generating interval and deviation, whereas with the burst traffic, the node generates double amount of data packet with the same interval and deviation. At the network layer, AREA-MAC examines three different types of simple yet effective routing schemes, namely N_0, N_1, and N_2 routing. With the N_0 routing, nodes simply broadcast a packet and any neighbor node irrespective of its location can receive and process the packet. With the N_1 routing, nodes send packets only to the up-level N_1 neighbors. Whereas, nodes with the N_2 routing send packets only to their up-level N_2 neighbors. The N_1 and N_2 up-levels of sensor nodes are explained in Chapter 6. We evaluate several metrics such as energy consumption, end-to-end delay, packet delivery ratio, and other of all three routing schemes.

Duty cycling of nodes makes the PHY layer very important player affecting considerably on the performance of a WSN MAC protocol. AREA-MAC periodically interacts with the PHY layer in order to distinguish between different radio modes and controls the radio efficiently by using appropriate channel, fading, and noise models.

4.3 Presentation of Results

Graphs presenting the evaluation results of AREA-MAC are drawn by using the *box-and-whisker diagram* for 11 different runs each for simulation and testbed experiments. A box-and-whisker diagram, shown in Figure 4.5, efficiently illustrates the distribution of a set of data based on the five-number summary: the smallest observation (minimum), lower quartile (Q1), median (Q2), upper quartile (Q3), and largest observation (maximum). Quartiles (percentiles) separate the data set into four equal parts, each representing one-fourth of the data. The upper quartile represents the highest 25% of the data, i.e., the median of the upper half, whereas the lower quartile represents the lowest 25% of the data, i.e., the median of the lower half.

The quartiles are also referred to as *hinges* and mark the ends of the box, whereas the *whiskers* stretch from the hinges to the extremes. If the median line inside the box is not equidistant from the hinges, the data is considered as *skewed* either at the right or left direction. The default box chart is generated using the

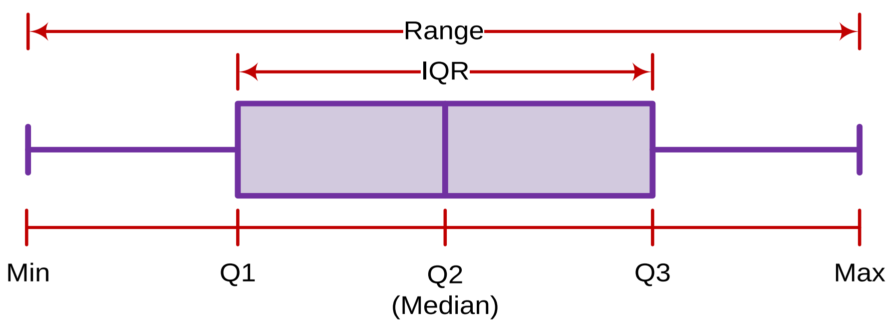


Figure 4.5: Box-and-Whisker Plot. The median is the middle value, half of the data set is below and half is above of it.

upper and lower quartiles, where the range between two middle quartiles is called an IQR . However, the box can be changed to one of other options such as standard deviation, standard error, a constant value, or a box representing the *confidence interval* about the median/mean value.

The rectangular box in each result graph of AREA-MAC protocol shows the confidence interval for $\alpha = 0.05$, i.e., with the confidence level of 95% about the mean value. The line within the box shows the median, while the upper and lower whisker bars show the maximum and minimum values respectively. All the median values in a graph have been connected with each other to elucidate their interrelation. Graphs have been scaled differently as per convenience.

Chapter 5

The IEEE 802.15.4

Don't judge each day by the harvest you reap but by the seeds that you plant.

–Robert Louis Stevenson

This chapter details the working of IEEE 802.15.4 with the special focus on its MAC protocol. After pointing out several limitations of the protocol, particularly related to energy, timeliness, and bandwidth factors, this chapter proposes a simple methodology to overcome most of the limitations. A network parameters setting to achieve an enhanced performance for an IEEE 802.15.4 based WSN has also been derived in this chapter.

5.1 Overview

Recent trends towards achieving short range and low power communication have laid down the foundation of WPANs. The IEEE 802.15 working group [115] deals with the development of consensus standards for WPANs in order to achieve the required functionalities while facilitating the coexistence of different portable and mobile computing devices. Diverse demands from consumers in terms of bandwidth, data rates, and QoS have resulted different 802.15 standards. Table 5.1 briefly compares the specifications, applications, and capabilities of different IEEE 802.15 WPAN standards.

The IEEE 802.15.4 [42] (referred to as 802.15.4 hereinafter) defines the PHY and MAC layers for an LR-WPAN. An LR-WPAN is a simple, low cost, low power, low QoS, and low data-rate communication network. It facilitates ease of installation, reasonable battery life, and reliable data transfer among devices within the limited range of around 10 meters. This protocol was not specifically designed for WSNs, but its capability to fit with different WSN requirements by appropriately tuning parameters has enabled it as a front runner for several WSN applications. In fact, its pertinence to WSNs has already been supported by several commercial

Table 5.1: IEEE 802.15 WPAN standards. Although each standard has targeted slightly different application domain, they all aim at reliable connectivity between portable and mobile devices in a close propinquity for better interoperability. 802.15.6 and 802.15.7 are currently at infancy stage and their architectures are being developed.

Standard	**Name**	**Data rate**	**Applications**	**QoS**
802.15.1	Bluetooth	1 Mbps	cell phones, laptops, PDAs, bar code readers, sensors, microphones	suitable for voice applications
802.15.2	coexistence of bluetooth and 802.11b	N/A	N/A	N/A
802.15.3	HR-WPAN	>20 Mbps	digital imaging and multimedia applications	very high QoS
802.15.4	LR-WPAN	<0.25 Mbps	industrial, agricultural, medical, surveillance, sensors, actuators with very low-power, low-cost	relaxed QoS and data rates
802.15.5	Mesh Networks	N/A	coverage extension, route redundancy, easier network configuration, better battery life	low to high (application dependent)
802.15.6	WBAN	<10 Mbps	nodes placed on or inside the human body, medical, sport/fitness, and entertainment fields	application dependent, mostly high QoS
802.15.7	Visual Light Communications (VLC)	10-100 Mbps	to support dual use of LED lighting, i.e., illumination and communication to enable ubiquitous computing	application dependent

sensor vendors. Combined with ZigBee [100], which provides the upper (network and application) layers, 802.15.4 defines a full protocol stack suitable for several WSN applications in surveillance, home automation, health care, industrial, and agricultural fields. Most of the new age sensor transceivers have built in compliance with 802.15.4. Some of the characteristics of 802.15.4 are as follows:

- Data rates from 20 to 250 kbps
- Star and peer-to-peer based network operations with FFDs and RFD
- Low power consumption with ED and LQI facilities
- Guaranteed Time Slots (GTSs) for low latency applications
- Low device, installation, and maintenance costs
- POS of 10 meters

- CSMA-CA channel access
- 16-bit short or 64-bit extended device addresses
- Fully acknowledged protocol for reliable data transfer
- 16 channels in the 2.4 GHz band, 30 channels in the 915 MHz band, and 3 channels in the 868 MHz band

5.2 An 802.15.4 LR-WPAN

An 802.15.4 LR-WPAN may consist of multiple FFDs and RFDs, with one of the FFDs promoted as the *PAN coordinator* to control the functionalities of the PAN. The coordinator associates/disassociates other FFDs and RFDs to the PAN and reserves resources for the associated devices. An FFD can communicate to RFDs or other FFDs, while an RFD can only communicate with an FFD. In order to avoid any Inter-PAN interference, each PAN chooses a unique identifier. A PAN can operate under a centralized *star topology* or a distributed *peer-to-peer* topology, as shown in Figure 5.1. In the star topology, a PAN coordinator controls the association and resource reservation for devices, which are only allowed to communicate with the PAN coordinator. The peer-to-peer topology also has a PAN coordinator to associate devices who are willing to join the PAN, however devices are independent and can communicate with each other. Devices in a peer-to-peer network can work in an ad hoc and multihop fashion by self-organizing themselves. Thus, this type of network can ideally represent a WSN application. 802.15.4, however, mainly concentrates on functionalities of the star topology with many specifications of the peer-to-peer scheme are left undefined [14].

An 802.15.4 LR-WPAN may operate in a *beacon-enabled* or a *nonbeacon-enabled* mode. With the beacon-enabled mode, the PAN coordinator periodically broad-

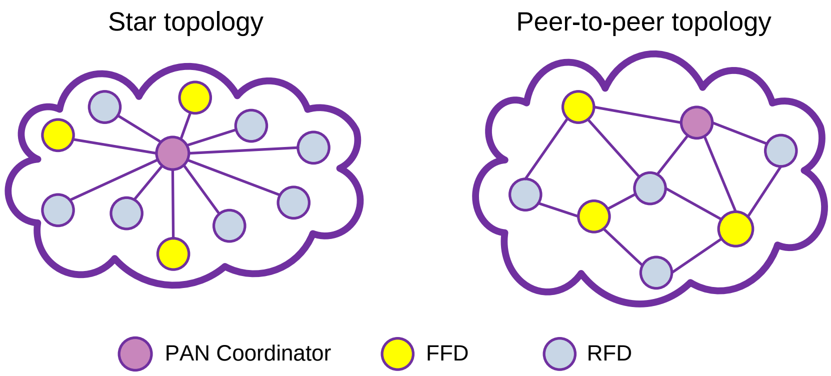


Figure 5.1: IEEE 802.15.4 Topology Structure. All devices of a PAN use unique 64-bit addresses to communicate with the PAN. Alternatively, the PAN coordinator may assign 16-bit short addresses to devices once they are associated with the PAN.

casts beacons and the associated devices follow and synchronize themselves to beacons. Alternatively, with the nonbeacon-enabled mode, the PAN coordinator normally does not transmit beacons unless it receives a beacon request from a device. The device may need a beacon for the purposes such as to scan or identify the existing PAN. To associate with a PAN, a device initially performs a channel scan to locate the PAN and its coordinator.

5.3 802.15.4 Architecture

The architecture of 802.15.4 targets the bottom two layers of the OSI communication model, i.e., the PHY and MAC layer, as shown in Figure 5.2. Both the layers are detailed below, while the specification and working of upper layers is defined by the ZigBee Alliance [100] and is out of our scope.

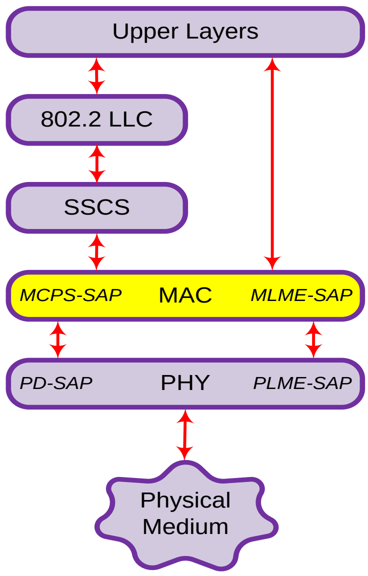


Figure 5.2: IEEE 802.15.4 protocol architecture. ZigBee targets the upper layers that usually include an application and a network layer. The application layer provides a user-specific intended function of the device and the network layer facilitates network configuration and message routing. An IEEE 802.2 Type 1 LLC can access the MAC sublayer through the SSCS.

5.3.1 PHY Layer

The PHY layer of 802.15.4 provides an interface between the MAC layer and the physical channel. It provides two services, each accessed through a SAP. The PHY data service is accessed through the PD-SAP, whereas the PHY management service is accessed through the PLME-SAP. The PHY data service enables the transmission and reception of PPDUs across the physical radio channel. The

PLME-SAP allows the transport of management commands between the MLME and PLME.

The PHY layer is responsible for activation and deactivation of the radio transceiver, ED, LQI for received packets, channel frequency selection, CCA, and transmitting as well as receiving packets across the physical medium. The radio operates at three different frequency bands with different modulation schemes and with the following possible combinations.

- An 868/915 MHz DSSS PHY employing BPSK modulation
- An 868/915 MHz DSSS PHY employing O-QPSK modulation
- An 868/915 MHz PSSS PHY employing BPSK and ASK modulation
- A 2.4 GHz DSSS PHY employing O-QPSK modulation

5.3.2 MAC Layer

The MAC sublayer offers an interface between the SSCS and the PHY layer. Like the PHY layer, it also provides two services: the MAC data service accessed through the MCPS-SAP and the MAC management service accessed through the MLME-SAP. The MCPS-SAP enables the transmission and reception of MPDUs across the PHY data service and also supports the transport of SPDUs between peer SSCS entities. The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. The main responsibilities of the MAC sublayer are beacon generation, management, and synchronization, channel accessing through CSMA, GTS management, frame validation and reliability, association/disassociation of devices, and supporting device security.

Devices of an 802.15.4 network mainly use the contention based CSMA-CA channel accessing method, however the PAN coordinator can allocate optional GTS time slots to devices that carry time critical traffic. With the nonbeacon-enabled mode, the MAC is ruled by an unslotted CSMA-CA, whereas with the beacon-enabled mode, it uses a slotted CSMA-CA. Note that ACK and beacon frames are transmitted without any contention. Under the unslotted CSMA-CA, a device, before a transmission, first waits for a random period. If the channel is found to be idle it transmits its data. Otherwise, the device waits for another random period before the next trial. With the slotted CSMA-CA, backoff slots are aligned with the start of the beacon transmission from the PAN coordinator. Before transmitting, a device locates the boundary of the next backoff slot and then waits for a random number of backoff slots. If the channel is still free, the device begins transmitting on the next available backoff slot boundary. Otherwise, it waits for another random number of backoff slots before trying to access the channel again.

The 802.15.4 MAC protocol can provide low duty cycle from 100% to 0.1%, enabling it as a worthy option for several WSN applications [116]. However, for energy efficient operations, the beacon-enabled network is needed so that synchronized sleep and wake-up mechanism can be adapted [117].

5.3.2.1 MAC Superframe Structure

With the beacon-enabled mode, the PAN coordinator defines a MAC superframe structure, drawn in Figure 5.3, by periodically transmitting beacons. The superframe comprises of an active part and an optional inactive part, where PAN devices may switch to low-power sleep mode. The active part of the superframe is divided into 16 equally sized slots. The beacon is always transmitted in the first slot and is used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframe. The active part of the superframe can be further subdivided into a CAP and a CFP. The CFP portion always appears at the end of the active part starting at a slot boundary immediately following the CAP.

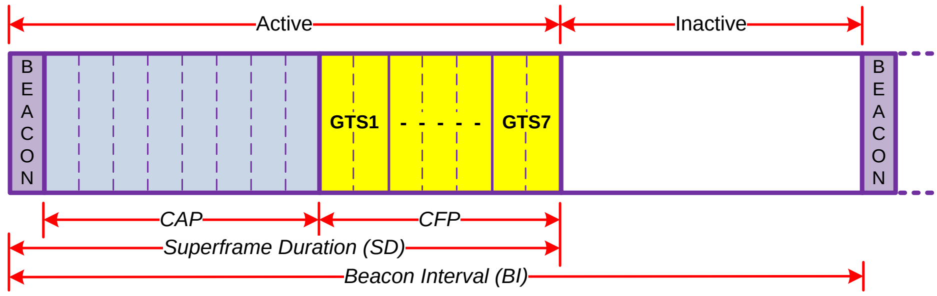


Figure 5.3: MAC superframe structure. It is divided into 16 equally sized slots and is bounded by network beacons. The superframe can have an active and an inactive portion. The CFP portion of the active part is used for low-latency applications.

A device that wishes to communicate during the CAP competes with other devices using a slotted CSMA-CA. For energy critical operations, the PAN coordinator can enable the *BatteryLifeExtension* (BLE) parameter. If this parameter is disabled, the associated device is enabled for the entire CAP period; otherwise it is enabled only for a *macBattLifeExtPeriods* number of backoff periods after the IFS following a beacon. The value of the *macBattLifeExtPeriods* is dependent on the PHY parameters and ranges from 6 to 41. For low-latency or bandwidth specific applications, the PAN coordinator may dedicate a portion of the CFP, called GTSs, to devices. The PAN coordinator may allocate up to seven GTSs to requesting devices subjected to the available superframe capacity, and a GTS may occupy more than one CFP slot. However, a sufficient portion of the CAP (at least of the *aMinCAPLength* size with the default value of 440 symbols) remains for contention based access of other devices or for new devices that wish to join the PAN. All contention based transactions are completed one IFS before the end of the CAP. Similarly, a GTS device completes its transmission one IFS before the time of the next GTS or the end of the CFP (in case if it has the last GTS in the current superframe).

The structure of the superframe is described by two values, namely the *macBeaconOrder* (BO) and the *macSuperframeOrder* (SO). BO describes the interval at which the coordinator transmits its beacon frames, i.e., the *Beacon Interval* (BI) as per following relation:

$$BI = aBaseSuperframeDuration \times 2^{BO} \qquad 0 \leq BO \leq 14 \qquad (5.1)$$

SO describes the length of the active portion of the superframe, i.e., the *Superframe Duration* (SD). The values of SO and SD are related as following:

$$SD = aBaseSuperframeDuration \times 2^{SO} \qquad 0 \leq SO \leq 14 \qquad (5.2)$$

Where $SO \leq BO$. The parameter *aBaseSuperframeDuration* is dependent on the data rate and the frequency range of the operation. For example, for the 2.4 GHz frequency band and 250 kbps data rate, 802.15.4 fixes the value of *aBaseSuperframeDuration* as 15.36 ms, which also denotes the minimum SD value corresponding to SO = 0. If BO = 15, the coordinator does not transmit beacon frames except when a device sends a beacon request. Similarly if SO = 15, the superframe does not remain active after the beacon.

5.4 Time, Energy, and Bandwidth Related Limitations of 802.15.4

Though an 802.15.4 network fulfills many of the WSN challenges, it still endures several limitations, especially for timeliness, energy, and bandwidth related issues, which are outlined in this section.

Note that SO and BO are the key parameters for potential energy savings for an 802.15.4 WSN. They calculate values of SD and BI parameters and define the inactive period of the superframe, where PAN devices can switch their radio to sleep mode. For time and bandwidth critical data, an associated device can request the PAN coordinator for GTS slot(s) by using a *GTS Characteristics* field. The format of the GTS Characteristics field, shown in Figure 5.4, is one byte long and contains three subfields. The *GTS Length* defines the number of GTS slots being requested by the device. The *GTS Direction* is defined relative to the data flow from the device that owns the GTS. It is a receive-only GTS if the GTS Direction

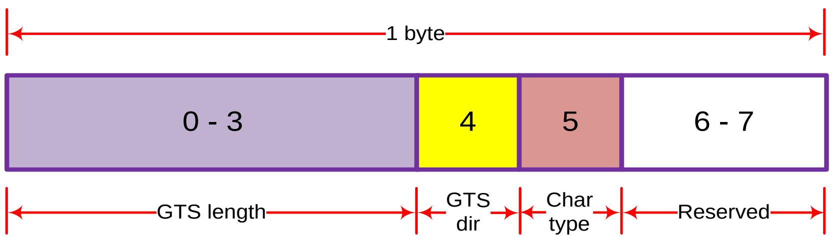


Figure 5.4: GTS Characteristics Field. The original GTS characteristics field consists of a 4-bit long GTS length for the number of GTS slots a device needs, a GTS direction bit to show whether it is a transmit- or receive-only GTS request, and a characteristics type bit to show whether it is an allocation or deallocation request.

is set to one and is a transmit-only GTS if the GTS Direction is set to zero. The *Characteristics Type* distinguishes whether the request is to allocate (value 1) or to deallocate (value 0) GTS slots.

If the GTS request is being accepted and acknowledged, the device distinguishes its GTS slot through a *GTS descriptor* field specified in the upcoming beacon. The format of the GTS descriptor is depicted in Figure 5.5.

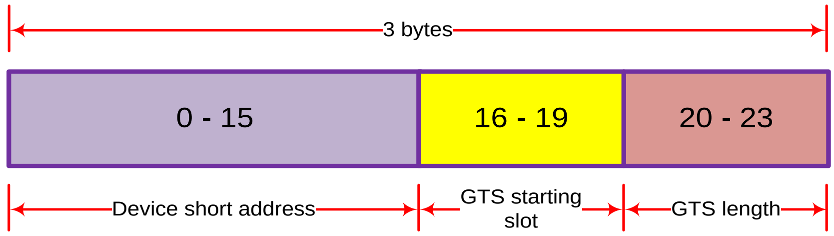


Figure 5.5: Format of the GTS Descriptor. It specifies the devices that have been allocated GTS slots with the information of their starting and ending GTS slots.

In order to efficiently utilize energy, time, and bandwidth related services provided by 802.15.4 MAC, following limitations - which are mostly related to the GTS service - need to be addressed:

- The first and foremost issue with the current GTS allocation mechanism provided by 802.15.4 is the bandwidth under-utilization. The standard only supports values of SD and BI by powers of two times a base constant (Equations (5.1) and (5.2)). The slot length, which is 1/16 of the SD value, grows exponentially with the increase in the SO value. A GTS also consists of an integer number of CFP slots. All these restrictions result in an inflexible GTS allocation for devices. Most of the time, a device uses/requires only a small portion of its allocated GTS(s), major portion may remain unused. It creates an empty hole in the CFP, like the memory fragmentation problem for operating systems.

- The protocol only supports an explicit GTS allocation for at most seven GTS requests in a superframe.

- 802.15.4 MAC only supports a first come first serve based GTS allocation and does not take into account the traffic, delay, and energy specifications for the requesting devices.

- A device requests for the required number of GTS(s) via GTS length subfield of the GTS characteristics field and the coordinator allocates the requested slots provided there is sufficient available capacity in the superframe. The device can request for all seven GTS slots at a time, even if it does not really need all of them. Since the coordinator does not use any check and balance system, it can allocate all GTS slots to the device. Such unbalanced slot distribution can block other needful devices carrying time bound traffic at least for the next beacon time.

- GTS slots allocated by the coordinator to PAN devices are expired on the basis of some constant factors. For a transmit-only GTS, the coordinator expires the GTS allocation of a device, if the data frame is not received from the device in the GTS at least every $2n$ superframes. Alternatively, for the receive-only GTS, the coordinator considers that a device is no longer using its GTS if an acknowledgement frame is not received from the device at least every $2n$ superframes. The value of n in both cases is given by:

$$n = \begin{cases} 2^{(8-BO)}, & 0 \leq BO \leq 7 \\ 1, & 8 \leq BO \leq 14 \end{cases} \tag{5.3}$$

  Moreover, the assigned GTS slots are broadcasted for the *aGTSDescPersistenceTime* (a constant having value of 4) number of superframes. These restrictions cause unnecessary energy consumption and GTS blockage for a longer period of time for other devices, if devices do not really use the allocated GTS(s) for the *aGTSDescPersistenceTime* number of superframe.

- Even if no device requests for a GTS slot and the actual CFP part is not present in the superframe, beacons transmitted by the PAN coordinator always use unnecessarily one byte for the CFP, which results in energy waste for all PAN devices.

- The superframe structure must contain at least an *aMinCAPLength* size CAP. That, on one hand, gives a chance to contention based PAN devices to communicate or to new devices to join the PAN. However for strict real-time applications, one may need a flexible sized CAP so that the CFP part can be stretched (and of course be shrunk) as per application requirements.

- GTS slots allocated to the PAN devices are applicable only in the upcoming beacon. It means that devices have to wait for the next beacon to use this guaranteed time service. The larger value of BI, where beacons will arrive after longer time periods, limits the advantages of the GTS facility provided by 802.15.4.

5.5 Enhancing 802.15.4 by Considering the Mentioned Limitations

The research study in the 802.15.4 domain attracts more attention of researchers. Most of their work, however, has been subjected to its feasibility and coexistence with other distributed networks, CSMA-CA, and other general performance criterion [118–120]. A relatively small amount of the literature studies the time and energy related issues of 802.15.4 [121–123].

An implicit GTS allocation scheme (i-GAME) for time-sensitive WSNs is proposed in [121], where the PAN coordinator tries to share the same GTS between multiple devices based on a *schedule*. A device that wants to use the GTS service sends its traffic and delay requirements to the coordinator. The coordinator, which runs an admission control algorithm, checks if a schedule satisfying requirements of the device is available. The coordinator approves the allocation request if such a schedule is available, otherwise rejects the request.

The Adaptive GTS Allocation (AGA) scheme for 802.15.4 [122] considers the past GTS usage of devices. The scheme is comprises of two phases namely the classification phase and the GTS scheduling phase. Devices are dynamically assigned priorities based on recent GTS usage feedbacks during the classification phase. Devices with higher priorities are assumed to have more recent and heavy traffic. The priority of a light traffic device is also gradually increased so that its starvation can be avoided. In the GTS scheduling phase, GTSs are allocated to devices in a non-decreasing order of their priorities.

The Multi-Beacon Superframe (MBS) algorithm [123] tries to decompose a superframe into two smaller superframes, which can be further decomposed into more smaller ones. Thus a single beacon interval is divided into multiple proper sub-beacon intervals, which increases the number of GTS slots for the master superframe and helps in reducing the bandwidth under-utilization problem. A device uses a Greedy GTS Allocation (GGA) algorithm and before joining a WPAN, it first decides based on its traffic characteristics which sub-beacon intervals are currently appropriate for it.

5.5.1 802.15.4 BED - A Proposed Scheme

The above proposed schemes mostly cause complexity and overhead for the PAN coordinator in assigning priorities, keeping previous GTS record of devices, decomposing a superframe into many smaller slots, sharing a GTS among multiple devices, and so on. As discussed in Chapter 3, the formation and maintenance of synchronization in WSNs is itself a very complex and resource and time consuming task. Therefore any extra complex responsibility will definitely manifold the troubles of a PAN coordinator. Furthermore, most of the 802.15.4 related research has been conducted on star topology based architecture, which seriously limits scalability and adaptability of an 802.15.4 network.

In this study, we aim to keep most of the 802.15.4 format intact and propose a simple scheme to overcome most of the mentioned limitations. We call this scheme the *802.15.4 BED*, where BED represents the Bandwidth, Energy, and Delay efficiency. This scheme targets to share responsibilities between a coordinator and devices rather them putting all the burden on the coordinator. Note that the main objective of this thesis work is to propose a novel energy and delay efficient AREA-MAC protocol for WSNs, which is promising to rectify the weakness of the other proposed MAC protocols in term of energy, delay, and other factors. Therefore, we do not intent to modify the basic structure of 802.15.4, particularly implications of the protocol in using a star based synchronized network.

The basic working of 802.15.4 BED scheme is depicted in Figure 5.6, where a coordinator periodically broadcasts beacons. All devices in the communication range of the coordinator can scan the PAN, synchronize with the coordinator, and receive beacons. The proposed scheme considers an 802.15.4 based WSN working in a beacon-enabled mode with a star topology. The WSN uses 2.4 GHz frequency band with O-QPSK modulation as this configuration provides an efficient network parameters setting [124]. The coordinator generates a beacon during the beacon generation phase. After that the CAP starts that follows the CFP. During the inactive period of the superframe, PAN members including the coordinator switch to sleep mode. During the beacon generation phase, the coordinator first collects data from its CAP regarding the devices that need GTS slots and then assembles and broadcasts the next beacon.

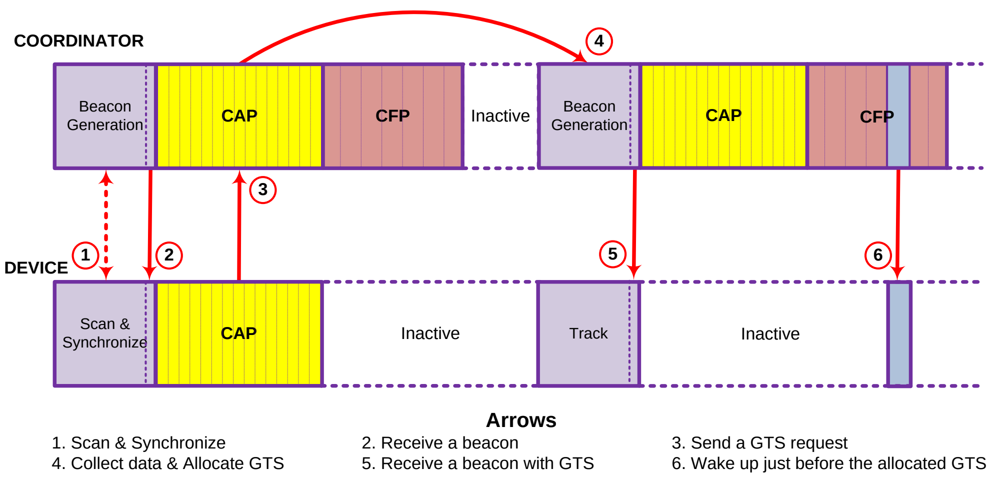


Figure 5.6: The 802.15.4 BED scheme. A device scans and synchronizes with the PAN and then receives a beacon. It may send a GTS request in the CAP and check whether the request has been accepted in the forthcoming beacon. To save energy, the device may enter sleep mode after receiving the beacon and wake up just before its GTS starts.

Alternatively, a device first scans the channel to locate the PAN and its coordinator and then synchronizes itself with the coordinator. It turns on its receiver just before the beacon arrival time and receives the beacon. In the following CAP, the device can request the coordinator for a GTS slot if it has delay bound data. Since the PAN coordinator does not initiate any data transfer and the GTS request of the device is only applicable to the next superframe, the device, following the GTS request, directly goes to the *early sleep mode* after accessing the CAP and prepares to wake up at the next beacon time in order to locate its GTS slots. Algorithm 5.1 depicts the periodic beacon broadcasting and receiving processes for the coordinator and the PAN devices respectively.

The device wakes up just before the next beacon arrival time and receives another beacon. It then deduces the fate of its previous GTS request by looking

Algorithm 5.1: startCycle()

```
if device == coordinator then                          /* for the coordinator */
    while not the end of experiment do
        broadcast beacon;
        collect GTS info;
        allocateGTS();
    end
else                                                   /* for a normal PAN device */
    if device is not associated then
        scan for a PAN;
        if PAN is found then
            synchronize with the coordinator;
        end
    end
    while not disassociated from the PAN do
        receive a beacon;
        request for GTS in CAP ; /* Request for GTS(s) if required */
        receive the next beacon;
        locateGTS();
    end
end
```

into the GTS Descriptor present in the beacon. If its GTS request has been accepted and is included in the GTS descriptor, it first goes to *early sleep mode* and wakes up just before its GTS starts. If the request was for a receive-only GTS, the device enables its receiver to receive data from the coordinator. Similarly the PAN coordinator enables its receiver to receive data from the device if the allocated GTS is a transmit-only. After using its GTS, the device switches back to sleep mode and prepares to receive the next beacon. Note that the device may also use the normal CAP along with its GTS slot as per requirement. After using the CAP, it goes to sleep mode providing the duration between its CAP access and its allocated GTS is greater than a threshold. The early sleep mode significantly reduces the energy consumption for the PAN devices. Algorithm 5.2 presents the working of a device in locating and using GTS slots.

As stated earlier, the beacon frame of the original standard holds at least one byte of the GTS field, even if no GTS slot is present. The coordinator of the 802.15.4 BED scheme, however, uses a reserved bit (*bit 13*) of the beacon frame to indicate the presence of its GTS fields, as shown in Figure 5.7. If there is any GTS allocation in the current beacon, the coordinator only then uses the GTS fields. Since beacons are periodically broadcasted and received by the coordinator and the PAN devices respectively, the use of the reserved bit to indicate the presence of GTS further saves the energy for the coordinator and the devices.

Algorithm 5.2: locateGTS()

read the GTS descriptor;
if *GTS request is accepted* **then**
  locate $myGTS$;
  if *wish to use CAP* **then**
    access CAP;
    if $myGTS - now > threshold$ **then**
      go to sleep;
      wake up just before $myGTS$;
    end
  end
  else
    go to sleep;
    wake up just before $myGTS$;
  end
  access $myGTS$;
end
else
  access CAP;
end
go to sleep;
prepare for the next cycle;

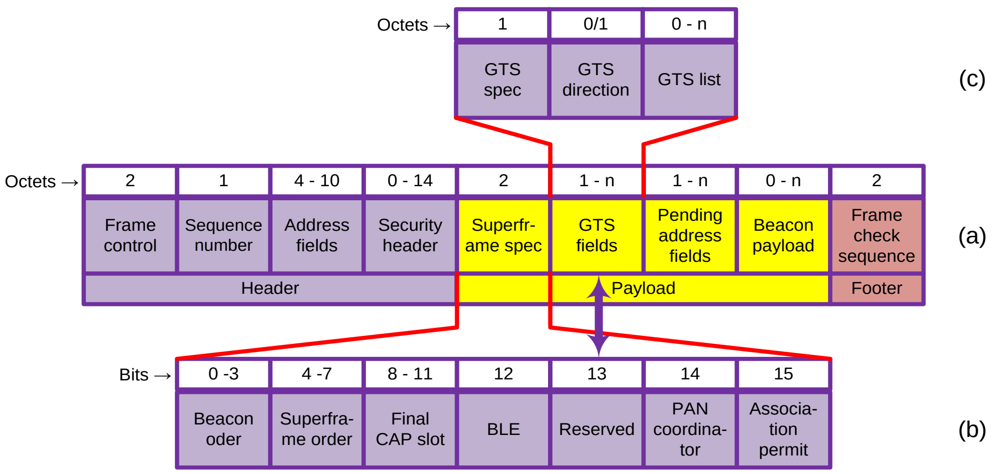


Figure 5.7: A beacon frame format of an 802.15.4 MAC protocol. The figure shows (a) the formats of beacon frame, (b) its superframe specification, and (c) its GTS fields. This format has been revised, where the reserved bit (bit 13) of the superframe specification has been used to present the GTS fields rather than using at least one byte of the beacon frame, even if no CFP slots are allocated to devices.

In order to address other limitations of 802.15.4, the proposed scheme uses a revised GTS characteristic field that is shown in Figure 5.8. A device, rather than blindly sending the requirement of a fixed length GTS, sends out its data length and delay specification to the coordinator. The coordinator selects an appropriate GTS length for the device as per its requirements and the remaining CFP size, and then allocates GTSs to devices in accordance with their delay conditions. In this way, the bandwidth under-utilization problem is avoided and the deserving devices get higher priority for the GTS allocation. To annul the constant GTS expiration, the coordinator uses the period bits of the revised GTS characteristics field and performs the GTS expiration dynamically. The device informs the coordinator via the period bits the number of times it needs GTS, which also frees the device from requesting the GTS again and again. The coordinator, however, can deallocate the GTS to a device if it needs GTSs for more deserving devices. Alternatively, the revised scheme can also be used helping the coordinator in assigning GTS on a round-robin or a time basis (for example, at every 5 seconds) to a device, once it declares to be using a low-latency based periodic traffic.

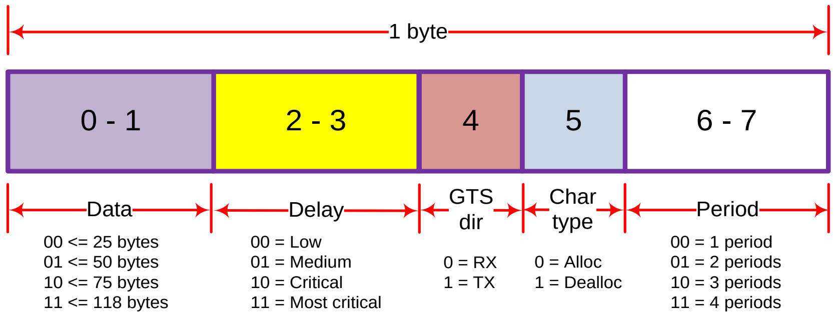


Figure 5.8: The revised GTS characteristics field. A device, rather than sending the GTS length directly, sends its data, delay, and period-cycle information to the coordinator, which then decides for appropriate GTS(s) for the device.

To nullify the restrictions of having at least an *aMinCAPLength* size CAP and having at most seven GTS in a superframe, the proposed scheme uses a simple alternate. If the coordinator observes that there are enough devices carrying critical delay bound data, it decreases the CAP and increases the CFP accordingly. If rarely the CFP part grows up to the full duration of the active part, the coordinator indicates devices by setting *final CAP slot* bits of the superframe specification to zero that there is no space for the CAP in the upcoming superframe. Devices that have not been allocated GTS immediately switch to sleep mode and prepare to receive the next beacon. In this way more devices carrying delay bound data are timely entertained and the energy consumption for non-GTS devices is also reduced. However, the coordinator ensures that such a zero-size CAP situation does not occur for two consecutive superframes, so that other contention based devices can avail their chances to communicate. Algorithm 5.3 explains the proposed GTS allocation scheme for the coordinator.

Algorithm 5.3: allocateGTS()

```
calculate GTS[n] for each requesting device;
for i ← 0 to 6 do
    allocate GTS[i] to a device in a delay order;
end
while devices with critical delay are unallocated do
    repeat
        decrease the CAP;
        i ← i + 1;
        allocate GTS[i] to the next needy device;
    until i < 15;
end
```

5.6 Selection and Implications of Different Parameters

In this section, we analytically examine the selection and implications of different parameters such as BO, BI, SO, SI, and CFP/GTS slots on the performance of an 802.15.4 based WSN. We also present a case study of how does the bandwidth under-utilization affect the GTS allocation and how does the 802.15.4 BED scheme improve it.

5.6.1 Relation between BO-BI and SO-SI

Values of BO and SO are important for the energy and latency related performance of a PAN as they decide how often beacons are transmitted and how long are their active and inactive periods. Figure 5.9 shows the relationship between BO and BI parameters based on Equation (5.1). The same relation is also valid between SO and SD parameters as per Equation (5.2). The value of BO (alternatively SO) lies between 0 and 14, whereas the value of BI (alternatively SD) varies between 15.36 ms and 251.6 s. We can choose an appropriate value of BI for the application. For example, for a periodic application, where devices generate/send data at an interval of 60 seconds to the coordinator, the application can choose 12 as the value of BI.

5.6.2 Maximum CFP Slots available

Before choosing an exact number of CFP slots that are needed by a device, we first calculate the maximum number of CFP slots available ($MaxCFP_{avail}$) for different values of SO. Given that the superframe is divided into 16 slots, a beacon is transmitted in the first slot, and the normal CAP should be at least of the size of *aMinCAPLength* that is equal to 440 symbols (1 slot = 60 symbols), the maximum number of available CFP slots for a given value of SO can be calculated by Equation (5.4).

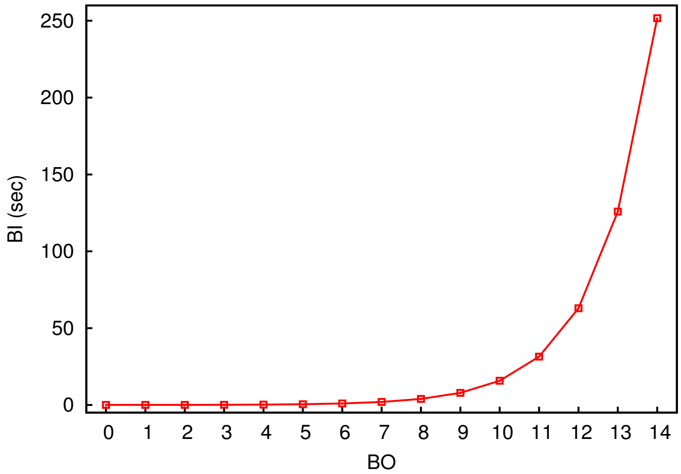


Figure 5.9: The relationship between the BO and BI parameters. Note that the same relationship is also valid between the SO and SD parameters.

$$MaxCFP_{avail} = \frac{15 - (aMinCAPLength/60)}{aBaseSlotDuration * 2^{SO}} \quad (5.4)$$

Based on Equation (5.4), Figure 5.10 presents the relationship between different values of SO and the corresponding maximum number of available CFP slots in a superframe.

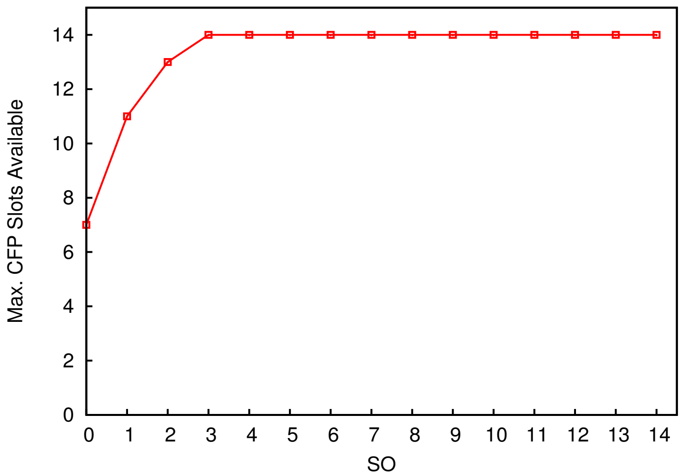


Figure 5.10: Maximum number of available CFP slots for a given value of superframe order. A coordinator can allocate 7 to 14 CFP slots as GTSs to PAN devices dependent on the SO value and the demand from devices.

5.6.3 CFP Slots Actually Needed

With the revised GTS characteristics field, the coordinator allocates GTS slots to a device depending on its delay and data specification. In order to calculate the number of slots needed for the device, we first calculate the total length of data packets it generates. The upper layers generate the data frame, which is passed to the MAC sublayer as a MSDU. 802.15.4 supports the maximum length of 118 bytes for a MSDU. The MAC sublayer adds a MHR of 9 bytes to the MSDU before converting it into a MPDU. The 127-bytes long MPDU is passed to the PHY layer as a PSDU. The PHY layer also adds a PHR of 6 bytes and converts it into a PPDU. Further 11 bytes are required for an (optional) acknowledgement request. To receive the acknowledgement, the device needs an *aTurnaroundTime* (12 symbols) to change the radio from transmit to receive (or vice-versa) mode.
Additionally, we have to consider the value of an IFS, which separates two successive frames sent by the device. Its length is dependent on the size of the frame that has just been transmitted. Frames, i.e., MPDUs of up to an *aMaxSIFSFrameSize* (18 bytes) long are followed by a SIFS period of at least a *macMinSIFSPeriod* (12 symbols) duration. Frames with lengths greater than an *aMaxSIFSFrameSize* are followed by a LIFS period of at least a *macMinLIFSPeriod* (40 symbols) duration; as formulated in Equation (5.5). Based on these calculations the number of CFP slots required for the device is derived in Equation (5.6).

$$IFS = \begin{cases} 12 & MPDU \leq 18\,Bytes \\ 40 & MPDU > 18\,Bytes \end{cases} \tag{5.5}$$

$$\begin{gathered} CFP_{req} = \frac{2 * Data\ + IFS + aTurnaroundTime}{aBaseSlotDuration * 2^{SO}} \\ where \quad Data = MSDU + MHR + PHR + ACK \end{gathered} \tag{5.6}$$

As an example, let's consider an application, where PAN devices generate data packets having length either 10, 25, 50, 75, 100, or 118 bytes. Based on Equation (5.6), we calculate the required CFP slots for a device with the acknowledgement transmission in Figure 5.11 and without the acknowledgment transmission in Figure 5.12. In both figures, we consider the value of SO as 0, 1, and 2. Both figures clarify that, even for the maximum size of data packet (118 bytes) and with the SO value of 2, devices need only one GTS slot in order to transmit their data. For the SO value of 1, devices need at most three CFP slots for the data transmission.

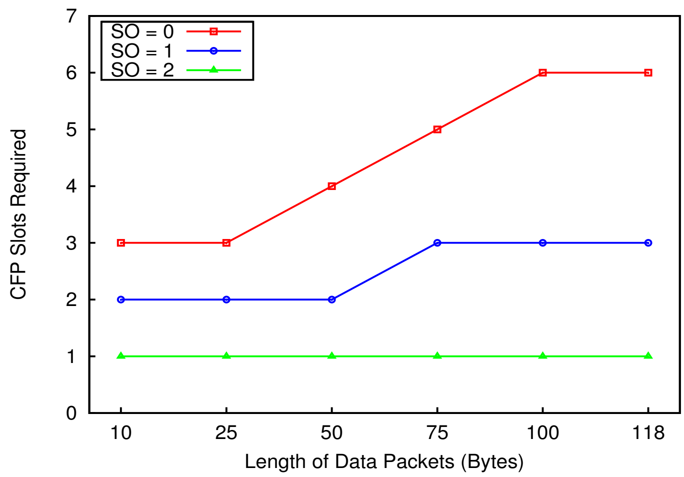


Figure 5.11: Length of data packets and required CFP slots with an acknowledgment transmission.

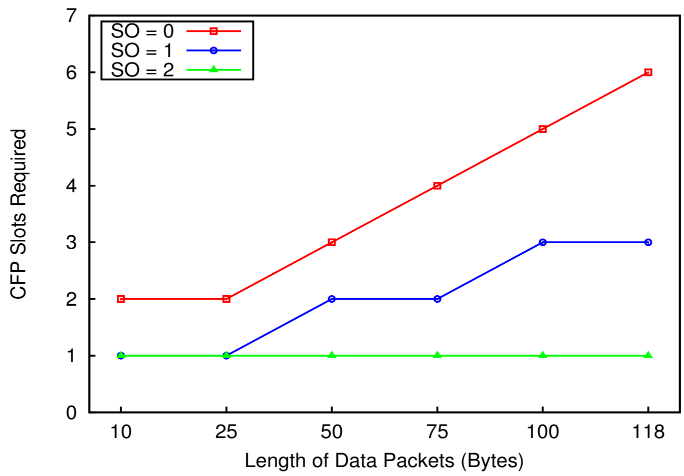


Figure 5.12: Length of data packets and required CFP slots without an acknowledgment transmission.

5.6.4 Bandwidth Under-Utilization

To verify the improvement offered by the proposed approach to the bandwidth under-utilization problem of 802.15.4, we consider an example of five different devices generating data packets of 100, 75, 50, 100, and 25 bytes and requesting for 3, 2, 1, 2, and 2 GTS respectively. With the original GTS allocation scheme, the coordinator would allocate up to at most seven GTS only to first three devices on the first come first serve basis and reject the remaining two devices. The rejected devices must wait for the next superframe, where they can retry to acquire the GTS. However, with the 802.15.4 BED scheme, where the coordinator calculates the appropriate number of CFP slots for a device as per its data size, all devices successfully acquire a GTS, as represented in Figure 5.13. Furthermore, two GTS are still unused that can be utilized by two more devices.

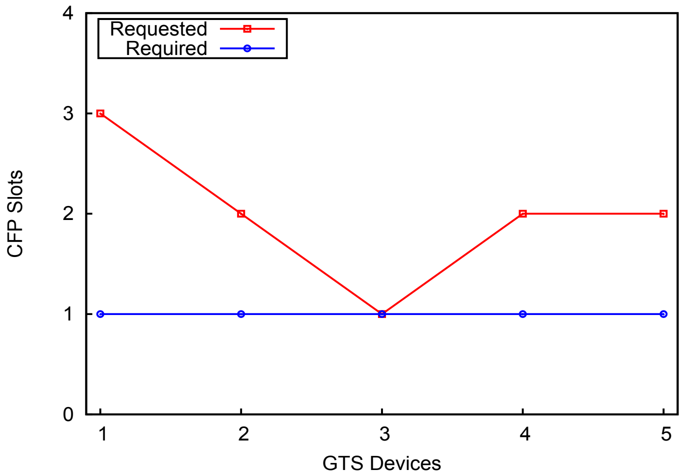


Figure 5.13: An example of bandwidth under-utilization. Five different devices request and acquire the differnt number of CFP slots for the length of data packets they generate. The original GTS allocation of the standard results in bandwidth under-utilization, as devices mostly need less than the requested or allocated slots.

5.7 Conclusions

In this chapter, we have elaborated the basic working of 802.15.4 with the special focus on its MAC protocol. Several limitations of the protocol, particularly related to energy, latency, and bandwidth factors have also been explored. Subsequently, we have proposed a simple scheme called 802.15.4 BED to cope with those limitations by presenting its basic working and the related analytical results. We have concluded that since the GTS length requested by devices mostly causes bandwidth under-utilization problem, the coordinator should assign GTSs depending on the data and delay constraints of devices. We have examined that even for the maximum packet length of 802.15.4 and with SO value of 2, a device needs only one GTS. Thus, the 802.15.4 BED scheme limits the bandwidth under-utilization problem and prioritizes the channel access of devices as per their delay requirements by helping the coordinator in allocating more GTS to more needy devices. Moreover, the proposed scheme significantly reduces energy consumption of PAN devices including the coordinator as compared to the default 802.15.4 standard.

Chapter 6

AREA-MAC Protocol

The horizon leans forward, offering you
space to place new steps of change.
–Maya Angelou

FROM now on, our work is dedicated to the design, implementation, and evaluation of AREA-MAC protocol. In this chapter, we discuss the basic design concepts and objectives of AREA-MAC and explain the ways it deals with energy, latency, redundancy, adaptivity, and other WSN challenges. At the end of the chapter, we analytically compare AREA-MAC with the protocols working on the traditional channel polling scheme and examine energy and delay gains of AREA-MAC.

6.1 Basic Concepts and Features

In Chapter 3, we have determined that the polling based channel accessing scheme conserves a substantial amount of node energy as compared to the contention (with common wake-up and sleep periods) and scheduling based schemes, which guides us to use this scheme for AREA-MAC protocol, which we propose in this chapter. However, the conventional channel polling scheme suffers from several limitations mainly causing from the use of long preambles that must be alleviated for energy and delay efficient operations of a WSN MAC protocol.

6.1.1 Limitations of the long preamble technique

Traditional channel polling based MAC protocols [46, 45, 99, 47] confront various limitations that are outlined below.

- *Excessive energy consumption for the sender as well as receiver*: Prior to any data transmission, a sending node sends out a long and extended preambles that is at least as long as the *check interval* of the intended receiver, and once the receiving node receives the preamble, it stays on and continues to listen until the data packet is arrived (Figure 3.9).

- *Overhearing at non-targeted nodes*: A neighboring node that senses the existence of a preamble on the channel stays on to receive the long preamble. Only after that it can detect that the data packet is not destined to it.
- *Excessive packet latency*: The transmission of a complete preamble before a data packet - irrespective to the point in time it is sensed by the intended receiver - increases the delay of each packet.
- *Higher collision probability*: Extended preambles cause for long transmission and reception durations for nodes that leads to a high collision probability.
- *Restriction of the duty cycle value*: Lowering the duty cycle extends the check interval duration of nodes, that is good from the receiver point of view, but it significantly increases the transmission cost and delay in the shape of long and extended preambles for a sender.
- *Longer channel utilization*: Being control packets, long preambles consume a higher percentage of the limited bandwidth available on the broadcast medium.
- *Non-compatibility with packet-based radios*: Byte-level radios can extend the preamble length as per check interval durations of nodes, however many advanced packet-based radios do not have the capability to stretch the preamble length beyond few bytes [16].

6.1.2 AREA-MAC Overview

The motivation behind designing AREA-MAC is to address the following issues: providing a suitable solution for both energy and delay efficient operations of a WSN by negating the adverse effects of the long preamble technique, offering coordinated operations without any synchronization overhead, reducing the number of unnecessary or duplicate packets, adapting gracefully to varying traffic and scaling conditions, and providing acceptable trades-offs between different parameters. The basic working of AREA-MAC is described below.

Nodes using AREA-MAC follow a duty cycle scheme and remain in *sleep mode* most of the time. They wake up shortly for a *wake-up* period at each *check interval* to sense the carrier. Figure 6.1 sketches the basic working of AREA-MAC and the relationship between sleep, wake-up, and check interval durations. Sleep and wake-up schedules of each node are unsynchronized and independent to that of other nodes. To poll or sample the channel, nodes use the channel polling access method and wake up shortly to check the availability of a preamble on the channel.

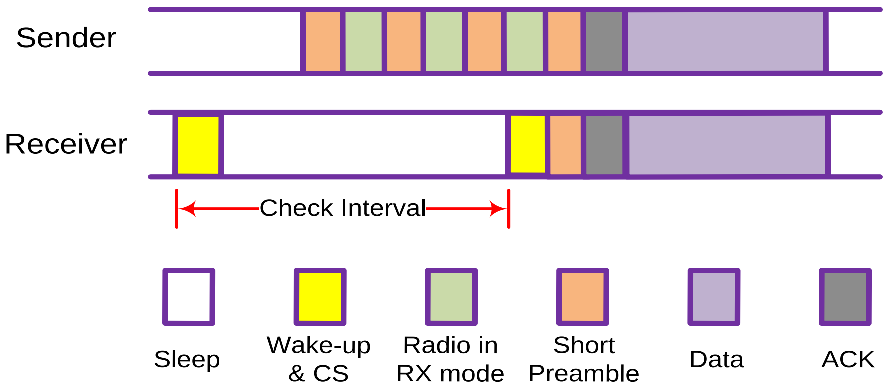


Figure 6.1: The basic working of AREA-MAC. This figure also shows the relationship between sleep, wake-up, and check interval periods.

To cope with the drawbacks of the conventional long preamble scheme, AREA-MAC uses a *short preamble* scheme, where nodes send out a stream of short preambles, instead of a long and fixed size preamble, prior to each data packet. Each preamble contains some additional information. Figure 6.2 shows the basic format of a 14-byte long preamble used by AREA-MAC nodes. The source and destination addresses embedded in the preamble enable nodes to send and receive an early response (we call it a *pre-ACK*) and to switch back to early sleep (referred to as *pre-sleep*) mode. The *sequence number* represents the unique identification number of the upcoming data packet, which is used to eliminate packet redundancy. The *message type* field distinguishes whether the upcoming data message is a *normal message* or a delay bound *real-time message*. The detailed bit length of each field of the preamble is further discussed in the respective simulation and testbed implementation chapters.

6.1.3 Transmission and Reception with AREA-MAC

Based on a duty cycle value, each node calculates its *burst count*, i.e., the maximum number of preambles to be sent within a given check interval duration. When the node has a data packet to send/forward, it starts with its *transmitting process* by sensing the carrier. On finding the carrier idle, the node transmits a burst of short preambles, each with a short spacing, so that the node can switch its radio to RX mode and can receive a pre-ACK from the intended receiver. The node

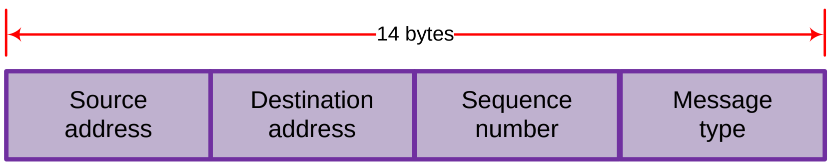


Figure 6.2: Basic format of an AREA-MAC preamble.

transmits a data packet as soon as it receives the pre-ACK frame. The sending node continues transmitting short preambles for a complete check interval duration unless it receives the pre-ACK frame from the intended receiver. The overall time a node spends for transmitting a data packet is denoted by the T_{tx}^{nr} (we call it the *normal processing* and will differentiate it with the *adaptive processing* later on) and is given by Equation (6.1). This value is the sum of times the node spends in carrier sense, radio switching, preamble transmitting, ack receiving, data packet transmitting, and bytes processing. Table 6.1 shows all the notations that will be used throughout the chapter.

$$T_{tx}^{nr} = T_{cs} + (n+3)(T_{sw}) + n(T_{tx}^{pr}) + T_{rx}^{ak} + T_{tx}^{dt} + (nL_{pr} + L_{dt} + L_{ak})T_{pr}^{bt} \quad (6.1)$$

The value of n lies between $0 \leq n \leq bc$ and represents the number of preambles sent by a node before it gets the pre-ACK from the receiver. The sending node switches its radio for $(n+3)$ times during this period as depicted in Figure 6.3: sleep→RX (to wake up and sense the carrier), RX→TX→RX (to start transmitting a preamble cycle containing n preambles and to receive a pre-ACK for every preamble), RX→TX (to send the data packet), and TX→sleep (to go back to the sleep mode). The sending node processes $(n+2)$ packets (n preambles, a data packet, and a pre-ACK frame) for a transmitting process. Figure 6.4 presents a simple state transition diagram for the node v for its transmitting process.

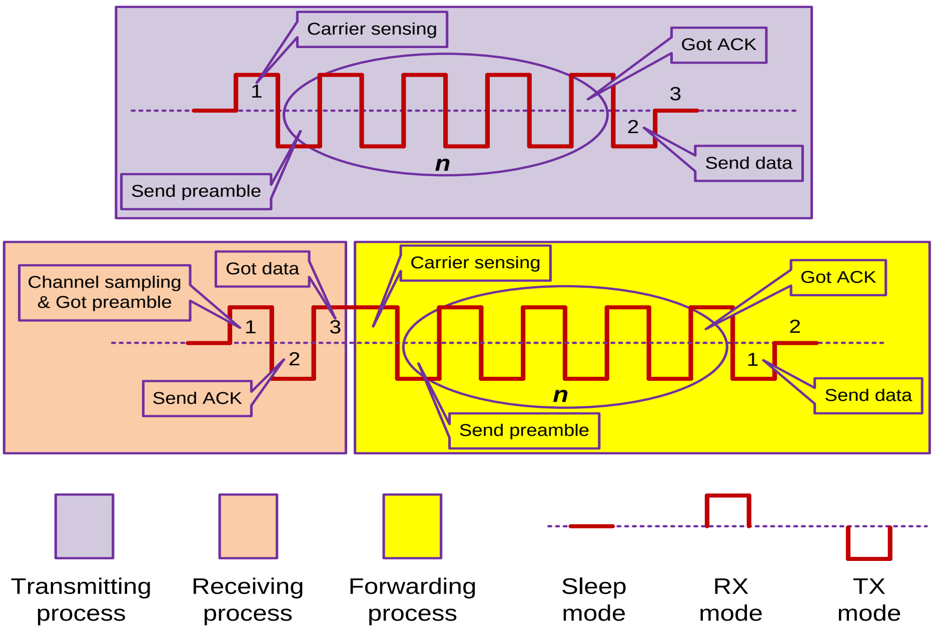


Figure 6.3: The radio switching between the transmit, receive, and sleep modes of an AREA-MAC node for its transmitting, receiving, and forwarding processes.

On the other side, a receiving node wakes up at every check interval duration and samples the channel to check the availability of a preamble. If the node finds

Table 6.1: Notations used throughout the chapter

Notation	**Used for**
bc	Burst count
T_{tx}^{nr}, T_{rx}^{nr}	Time in a transmit & receive (normal) process
T_{tx}^{ad}, T_{rx}^{ad}	Time in a transmit & receive (adaptive) process
T_{fd}^{nr}, T_{fd}^{ad}	Time in a normal & adaptive forward processes
T_{tx}^{pr}, T_{rx}^{pr}	Time in sending & receiving a preamble
T_{tx}^{ak}, T_{rx}^{ak}	Time in sending & receiving an ACK
$T_{tx}^{dt}, T_{rxt}^{dt}$	Time in sending & receiving a data packet
T_{tr}	Time in a treating process
T_{wk}	Time in a wake-up mode
I_{ci}	Check interval
I_{dt}	Data generating interval
E_t	Total energy consumption per unit time
E_{sa}, T_{sa}, P_{sa}	Energy, Time, & Power in sampling the channel
E_{cs}, T_{cs}, P_{cs}	Energy, Time, & Power in sensing the carrier
E_{sl}, T_{sl}, P_{sl}	Energy, Time, & Power in sleep mode
E_{rx}, T_{rx}, P_{rx}	Energy, Time, & Power in RX mode
E_{tx}, T_{tx}, P_{tx}	Energy, Time, & Power in TX mode
E_{sw}, T_{sw}, P_{sw}	Energy, Time, & Power in radio switching
E_{pr}, T_{pr}, P_{pr}	Energy, Time, & Power in processing a frame
T_{sa}^{av}, T_{cs}^{av}	Average time in sampling & sensing the carrier
T_{pr}^{bt}	Processing time for one byte
T_{bo}	Back-off time
T_{st}	Time for a contention slot
S_{cw}	Size of the contention window
L_{pr}, L_{dt}, L_{ak}	Length of preamble, data, & ACK frame in bytes
T_{bt}	Time to transmit/receive a byte
N	Total number of sensor nodes
$N_{v_i}^1$	1-level neighbors for v_i
$N_{v_i}^2$	2-level neighbors for v_i
δ_{v_i}	Total number of up-level neighbors for v_i
∂_{v_i,v_j}	Euclidean distance between v_i and v_j
C_{v_i,v_j}	Connectivity between v_i and v_j
t, τ	Data generating and deviation intervals

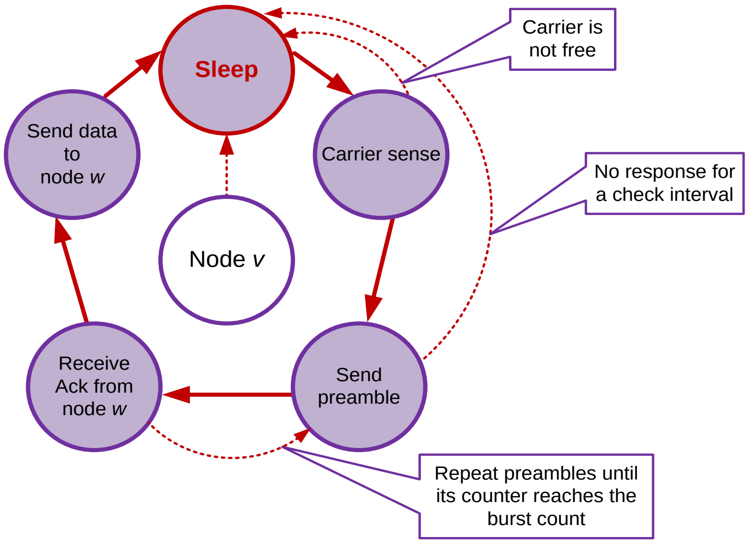


Figure 6.4: A state transition diagram for the sending node v when it sends a data packet to node w.

a preamble on the channel, it starts with its *receiving process*. If the destination address of the preamble matches with the address of the node, it immediately sends a pre-ACK frame to the sender, and then switches radio to the RX mode to receive a data packet from the sender. The node immediately goes back to the *pre-sleep* mode, if a preamble is not found or its destination address does not match with the address of the node. However, for broadcast communication, the node sends a pre-ACK frame as soon as it receives the preamble without checking the destination address. The node switches its radio three times during a receiving process, as sketched in Figure 6.3: sleep→RX (to wake up and sample the channel), RX→TX (to send a pre-ACK), and TX→RX (to receive a data packet). After receiving the data packet, if the receiving node is not the sink node, it immediately forwards the packet to the sink node by initiating a *forwarding process*.

A check interval duration of a node consists of sleep and wake-up periods. Unlike many other protocols, AREA-MAC does not consider a fixed size wake-up period, but adapts its size as per traffic conditions. A check interval duration of a node is given by:

$$I_{ci} = T_{wk} + T_{sl} \tag{6.2}$$

Where

$$T_{wk} = \begin{cases} 2T_{sw} + T_{sa}, & \text{if no preamble is found} \\ T_{tr}, & \text{if a treatable preamble is found} \end{cases} \tag{6.3}$$

$$T_{tr} = \begin{cases} 2T_{sw} + T_{sa} + T_{rx}^{pr} + L_{pr}T_{pr}^{bt}, & \text{if preamble is not for me} \\ T_{rx}^{nr} + T_{fd}^{nr}, & \text{else receive \& forward (a normal process)} \end{cases} \tag{6.4}$$

$$T_{rx}^{nr} = T_{sa} + T_{rx}^{pr} + T_{tx}^{ak} + T_{rx}^{dt} + 3T_{sw} + (L_{pr} + L_{dt} + L_{ak})\, T_{pr}^{bt} \tag{6.5}$$

$$T_{fd}^{nr} = T_{cs} + n\,(T_{tx}^{pr}) + (n+2)\,(T_{sw}) + T_{rx}^{ak} + T_{tx}^{dt} + (nL_{pr} + L_{dt} + L_{ak})\, T_{pr}^{bt} \tag{6.6}$$

Like the transmitting process, a node sends out a bunch of n short preambles unless it receives a pre-ACK, where $0 \le n \le bc$, for the forwarding process. On reception of the pre-ACK frame, the node forwards the data packet. The radio transition during the forwarding process is given in Figure 6.3, while Figure 6.5 presents a simple state transition diagram for the receiving and forwarding processes of node v.

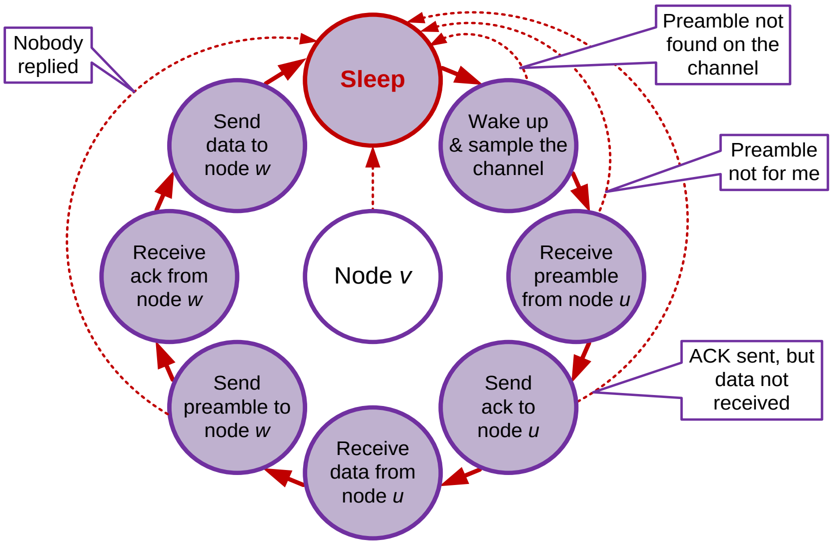


Figure 6.5: A state transition diagram for the receiving and forwarding processes of node v, where it receives a data packet from node u and forwards it to node w.

Figure 6.6 depicts a detailed flow chart sketching the transmitting and receiving processes of an AREA-MAC node. The undiscussed features shown in the flowchart are explained in the following sections.

6.1.4 AREA-MAC Features

To be able to carry through the unique challenges of WSNs, AREA-MAC offers several features that are briefed below. Some of them are detailed as per their relevance later on.

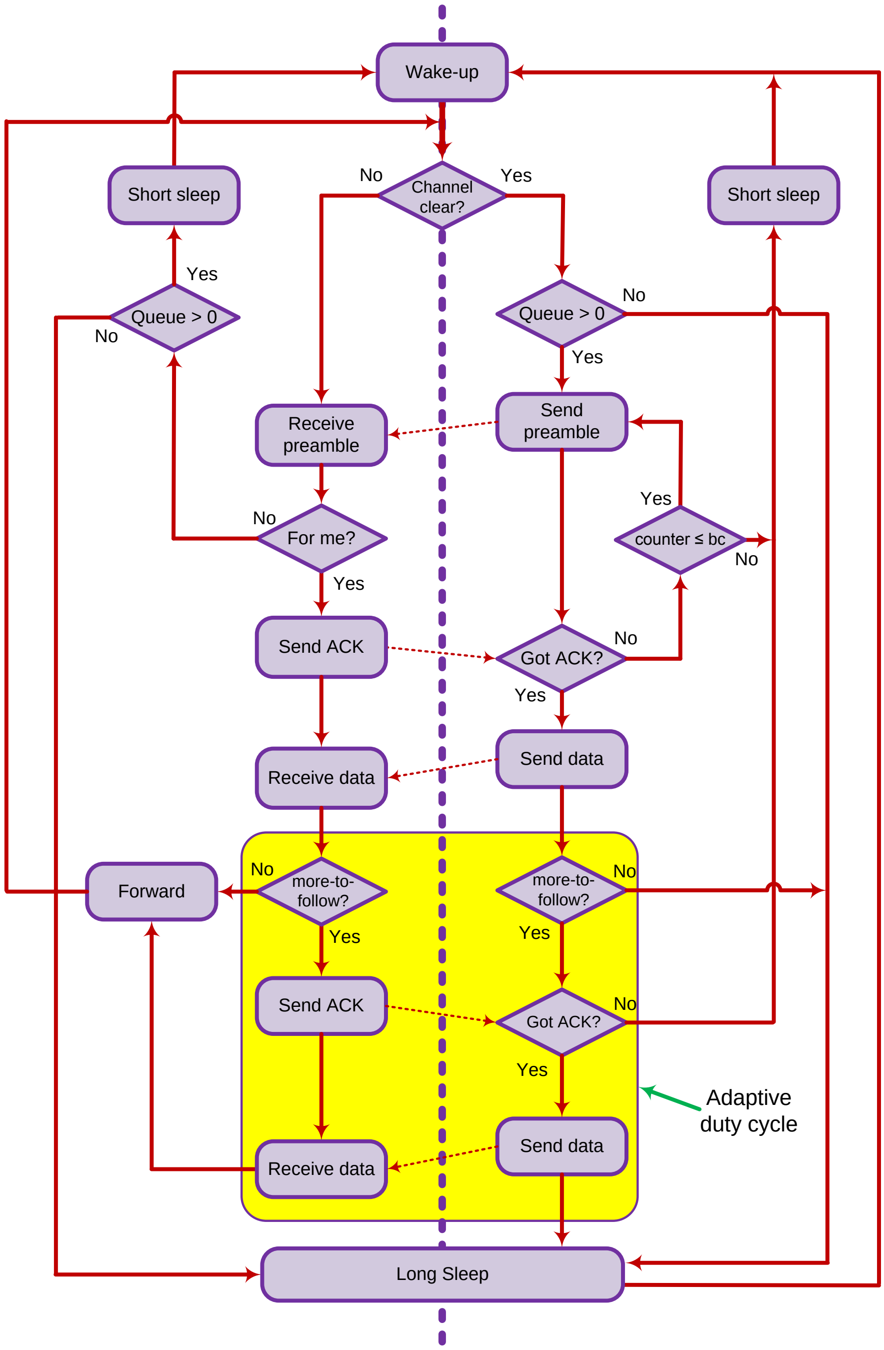


Figure 6.6: A flow chart showing the details of reception and transmission processes (left and right side of the dotted vertical line respectively) for a node using AREA-MAC.

6.1.4.1 Energy Efficient Design Strategies

In order to design an energy efficient MAC protocol for WSNs, the major sources of energy waste need to be addressed. As discussed in Section 2.4.2, idle listening, collisions, overhearing, over-emitting, and control overhead are the main perpetrators that drain most of the node energy. AREA-MAC deals with these factors as follows. Note that these factors are mostly inter-related and reducing one factor can affect others either in a positive or a negative way.

Coping with Idle Listening: Since a WSN usually generates much less data traffic and sends very small data frames comparing to other distributed networks, sensor nodes remain idle most of the time. The radio of a node may consume as much energy during idle listening as in receiving packets making idle listening one of the major sources of energy dissipate in WSNs. AREA-MAC deals with idle listening by the following ways:

- Nodes perform duty cycling, remain in sleep mode most of the time, and wake up shortly just to check the availability of a preamble on the medium.
- Nodes use the channel polling scheme to access the channel that is about 10 times less expensive (in terms of energy and time) than listening for a full contention period [87].

Coping with Overhearing: The pre-ACK and pre-sleep features of AREA-MAC considerably decrease overhearing for nodes in the following ways:

- Nodes send a burst (of at most the *burst count* size) of short preambles as opposed to the traditional long preamble method, where a long preamble is sent out. Short preambles integrated with source and destination addresses have threefold advantages:
  - A sending node, on receiving a pre-ACK, cuts back the preamble duration and immediately sends out a data packet.
  - The intended receiver shows its willingness to receive the packet by sending an early pre-ACK frame.
  - An overhearing neighbor immediately goes back to the sleep mode once it recognizes that the preamble (and the upcoming data packet) is not destined to it. It is unlike the long preamble method, where a node first receives the complete long preamble, and only then can distinguish whether it is the correct node the data packet is destined to.

Coping with Collisions: Both pre-ACK and pre-sleep options force nodes either to receive packets or to switch back to the sleep mode quite earlier as compared to the traditional long preamble scheme. This reduces the collision chances as the preamble transmission and reception durations are trimmed down. Moreover, the pre-ACK creates one sort of a *pre-established* communication path between the sender and receiver, where they can send and receive data packets relatively in a safe and reliable manner. This minimizes the chances of the data packet being dropped as the probability of data collision decreases. However, collisions might occur with preambles, which is not the severe case, as a node continues sending short preambles for a check interval duration unless it receives the pre-ACK.

Coping with Over-Emitting: Over-emitting or deafness occurs in WSNs when a node transmits packets while the intended receiver is still in sleep mode. Over-emitting may occur with AREA-MAC, especially during preamble transmission, as wake-up periods of nodes are designed to be shorter (in order to reduce idle listening and overhearing). Over-emitting is, however, almost eliminated during the actual data transmission, where both the sender and receiver communicate on a so called pre-established link. Over-emitting with AREA-MAC is further improved with the *adaptive duty cycling* feature, which is further discussed in Section 6.1.4.2. While sending a data packet, a node that still have more data packets in its queue waiting to be sent out indicates the receiving node by enabling the *more-to-follow* bit of the data packet. The receiving node, on finding the more-to-follow bits of the packet enabled, adapts its next wake-up schedule in accordance with the sending node, which decreases over-emitting for both preamble and data transmission.

Coping with Control Overhead: Since AREA-MAC does not require any sort of synchronization, nodes do not send and receive periodic control packets to keep themselves synchronized, which substantially reduces the control overhead for nodes. Adding extra information to each preamble, which is a control packet itself, may cause overhead for AREA-MAC nodes. Nevertheless, the smaller size of a preamble (a preamble is roughly one third of the data packet), pre-ACK, pre-sleep, adaptive duty cycling, and reduced redundancy (discussed later in the chapter) options significantly ease up the overhead complexity for AREA-MAC nodes.

6.1.4.2 Timeliness Support

Since one of the design objectives of AREA-MAC is to reduce packet latency for sensor nodes, it achieves this target in the following ways:

- The pre-ACK enables the source node to release the data packet and the receiving node to capture (and forward, if it is not the sink node) the packet in a timely manner. This is opposite to the long preamble technique, where a sender sends out a data packet only after transmitting a long and full preamble. The intended receiver that may sense the existence of preamble earlier has to wait for the completion of the preamble transmission.

- The adaptive duty cycle feature of AREA-MAC enables a receiving node to adjust its duty cycle in accordance with the sending node. The receiving node, on reception of a data packet with more-to-follow bit enabled, directly sends out a pre-ACK to the sender. This not only confirms the reception of the previous data packet, but also shows the readiness of the receiver to receive another data packet without the need of any preamble transmission. The sending node after receiving a second pre-ACK from the receiver, transmits another data packet without sensing the carrier. After receiving both packets, the receiving node forwards them to the next hop neighbor in the same way by enabling more-to-follow bit of the first data packet. This adaptive duty cycling has manifold advantages:

  - it reduces the packet latency (at least for the second packet).
  - it reduces the number of preambles to be transmitted for the sender and to be received by the receiver.
  - it reduces overhearing for the neighboring nodes.
  - it increases reliability of the data packet (at least for the packet that has more-to-follow bit enabled).

By default, AREA-MAC supports transmitting only two consecutive data packets via an adaptive duty cycling process. This number can, however, be increased, which on one hand would decrease the energy consumption of nodes, but on the other hand increases the queue size of the receiving node and blocks other neighbors for a longer period from accessing the channel. The receiving node needs longer to forward each of the queued packets, which increases their overall latency. Since a WSN generates much less data and an AREA-MAC node sends (or forwards) a data packet as soon as it generates (or receives) it, we assume that it would be a rare case when a node has more than two packets saved in its queue.

Figure 6.7 explains the adaptive duty cycle feature of AREA-MAC by indicating three different scenarios. First two scenarios for a receiving process are already mentioned in Equation (6.4), which can be rewritten for the third scenario as following:

$$T_{tr} = \begin{cases} 2T_{sw} + T_{sa} + T_{rx}^{pr} + L_{pr}T_{pr}^{bt}, & \text{if preamble is not for me} \\ T_{rx}^{nr} + T_{fd}^{nr}, & \text{receive \& forward (normal process)} \\ T_{rx}^{nr} + \Delta + T_{fd}^{nr} + \Phi, & \text{receive \& forward (adaptive process)} \end{cases} \tag{6.7}$$

Where

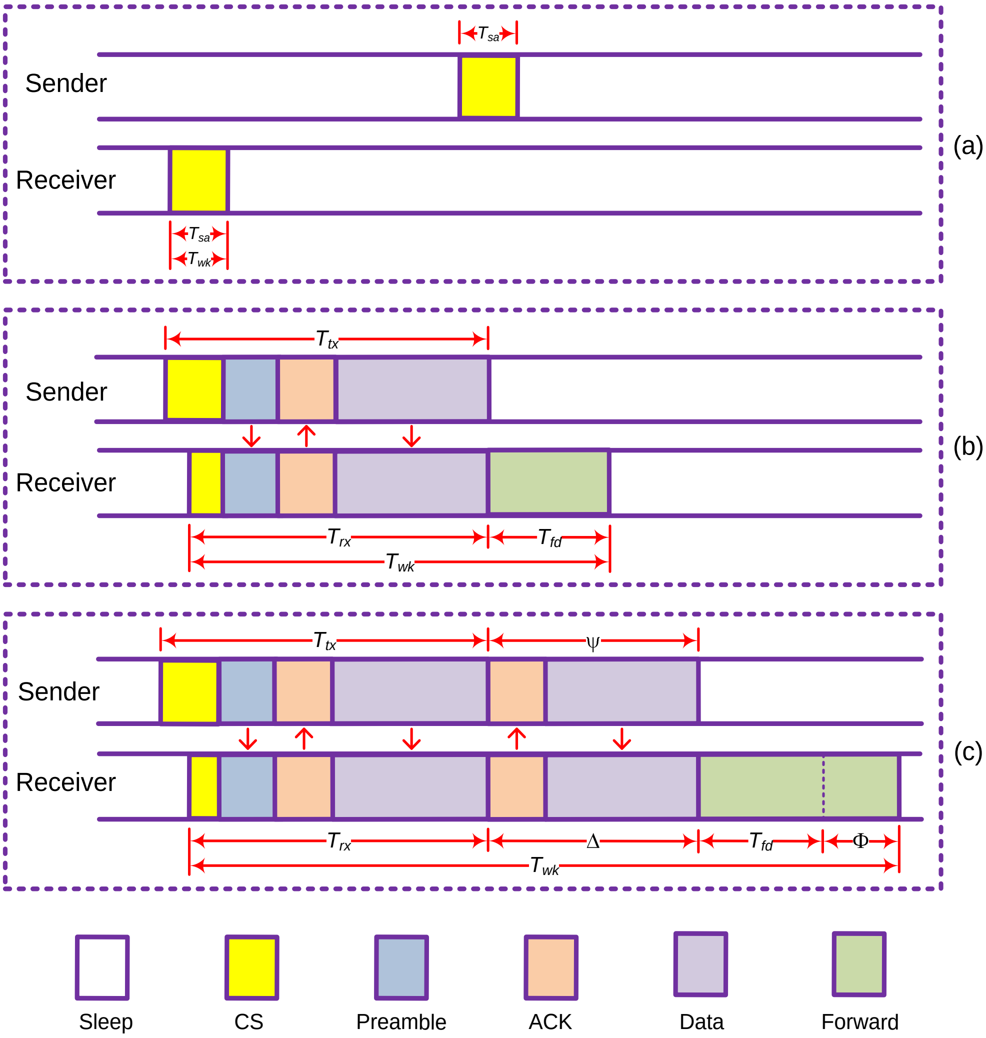


Figure 6.7: Adaptive Duty Cycling with AREA-MAC. A node adapts its duty cycle depending on the availability of the preamble/data packet on the channel. (a) the node does not find a preamble on the channel and immediately goes back to sleep mode, (b) it receives and forwards a data packet, i.e., a normal processing, and (c) it receives and forwards two data packets by adapting its wake-up period, i.e., an adaptive process.

$$\Delta = T_{rx}^{ad} = T_{tx}^{ak} + T_{rx}^{dt} + 2T_{sw} + (L_{dt} + L_{ak})\, T_{pr}^{bt} \tag{6.8}$$

$$\Phi = T_{fd}^{ad} = T_{rx}^{ak} + T_{tx}^{dt} + 2T_{sw} + (L_{dt} + L_{ak})\, T_{pr}^{bt} \tag{6.9}$$

Similarly, the sending node extends its transmitting process given in Equation (6.1) by Ψ for an adaptive processing as follows:

$$T_{tx}^{ad} = T_{tx}^{nr} + \Psi \tag{6.10}$$

Where

$$\Psi = T_{rx}^{ak} + T_{tx}^{dt} + 2T_{sw} + (L_{dt} + L_{ak})\, T_{pr}^{bt} \tag{6.11}$$

- A duty cycled node after waking up and performing a task (if there is any) goes back to the normal sleep mode. We call it a *long sleep* as the node wakes up again only after a complete check interval duration. Alternatively, while transmitting a data packet, if the node does not find the channel idle after all the backoff attempts, it goes to a *short sleep* mode and wakes up rather quickly to continue with its packet transmission. The short sleep not only keeps the packet latency minimum, it also provides a short space to neighbors to finish their current transmission without any collision. The value of the short sleep duration can either be chosen relative to the long sleep duration, i.e., a fraction of the duty cycle value, or it can be a fixed value.

- AREA-MAC considers two types of data messages, i.e., a normal message and a real-time message, to support both normal and real-time traffic for a WSN. The type of a message is indicated by the binary *message type* subfield of the preamble, where the zero value represents the normal traffic and the one represents the real-time traffic. After transmitting a preamble and during the waiting time to receive the pre-ACK, if the sending node receives a real-time preamble from its neighbor, it drops its bid of accessing the channel and starts responding the sending node (if the preamble is either of the broadcast type or destined to this node). Similarly, a sending node prefers to send out a real-time packet (if there is any) than a normal packet from its queue. Thus, a real-time packet gets a higher priority over the normal one.

- AREA-MAC mainly supports converge-cast communication, where nodes send/forward packets in all-to-one manner, i.e., towards the sink node. In order to further reduce packet latency, AREA-MAC supports three types of simple yet effective routing schemes that are explained in Chapter 7.

6.1.4.3 Decreasing Redundancy

Redundancy causes unnecessary energy consumption for nodes in processing duplicate packets. To avoid this, nodes using AREA-MAC assign a unique sequence number to each generated data packet, and place the sequence number of the upcoming data packet in each preamble. AREA-MAC assigns each node a unique ID (starting from 1 to n, where n is the maximum number of nodes) at the start-up. A node initializes its sequence counter with $(ID \times 1000)$ value, which is increased every time the node generates a packet. The node reinitializes the sequence counter once the counter reaches $(ID \times 1000 + 999)$ value.

On receiving a preamble, a receiving node first checks the sequence number into a list containing the sequence numbers of last x treated packets. If the packet has already been treated, the receiving node, if it is an *up-level neighbor* of the sender, sends a negative ACK ($NACK$) to the sender, which then discards the packet from its queue in order to reduce the redundancy. The description of an up-level neighbor is provided in Section 6.2.3.

6.1.4.4 Other Features

Next to the discussed features, AREA-MAC offers several other characteristics, which are concisely outlined below and are detailed throughout this thesis as per their relevance.

AREA-MAC implements fully independent sleep and wake-up schedules for each node and does not need any sort of synchronization among them. Nodes, however, *asynchronously coordinate* their transmission via more-to-follow and message type bits, and for that they do not need any sort of synchronization. Thus AREA-MAC is a self-organizing MAC protocol that does not need a central node or a cluster head to command the operations of nodes. Freeing nodes from the synchronization overhead also greatly helps a WSN to scale smoothly. Unlike cluster based approaches, where nodes only communicate via cluster heads, nodes using AREA-MAC are free to communicate directly with their peers. The more-to-follow and message type bits also help nodes adapting to traffic conditions. Additionally, in order to examine the effect of different network and traffic conditions, AREA-MAC offers support for different type of traffic generating scenarios under different topologies.

The communication between the sender and receiver for data packets on a pre-established track via the pre-ACK method helps in increasing the number of data packets that successfully arrive at the sink node. Thus, AREA-MAC increases the packet delivery ratio. Moreover, it supports both broadcast and unicast form of communications and also investigates the effect of the cross-layering by supporting different simple routing and data generating schemes at the network and application layer respectively.

6.2 AREA-MAC Design

Having discussed the basic concepts and features of AREA-MAC protocol, we, in this section, present the overall design of AREA-MAC by demonstrating its design phases, assumptions, network, energy, delay, and traffic models.

6.2.1 Design Phases

The design of AREA-MAC can be categorized into the following three major phases.

6.2.1.1 Network Setup Phase

During the network setup phase, each node calculates its burst count value depending on the provided duty cycle value. It also collects information about its node ID, location, and up-level and low-level neighbors and selects a random sleep and wake-up schedule to perform duty cycling. Each node also selects a random time to generate first data packet as per given data generating interval and deviation rates. The network setup phase lasts for a T_{setup} duration. We assume that T_{setup} is long enough, so that every node collects and calculates all the required information. As network setup phase occurs once, this consideration does not have any significant effect on the overall network performance.

6.2.1.2 Duty Cycle Phase

After collecting the required information, each node starts with the duty cycle phase, where it regularly switches the radio between sleep and wake-up modes. The node wakes up the radio at every I_{ci} duration to sample the channel for the availability of a preamble. On finding the preamble, the node initiates its receiving process follows by the forwarding process. Similarly, a node generates a random data packet at I_{dt} interval and sends it to the sink node, possibly via multi-hop communication. Algorithms 6.1, 6.2, and 6.3 reflect the working of a node during the duty cycle phase.

6.2.1.3 Stats Collection Phase

After performing the duty cycle processing for a given experiment runtime, all nodes collect statistics just before the end of an experiment. The statistical data includes the number of data packet generated/sent/forwarded, the number of preambles sent/received, the number of overheard packets received, total radio time spent in the receive/transmit/CS/sleep modes, and end-to-end delay for a packet. All the statistical data are stored for further analyzation.

6.2.2 Assumptions

We consider a WSN consisting of several nodes and a sink node. The function of the sink node is to receive data packets from other nodes and to forward them to the end-user/end-terminal. All other nodes are normal nodes performing the functions of generating, receiving, and transmitting packets. The radio of a node operates in four different modes, namely TX mode, RX mode, Listen (CS and/or channel sampling) mode, and sleep mode. We assume that all nodes are stationary

Algorithm 6.1: wakeupProcess()

```
if now == I_dt then                                    /* data generating interval */
    generate a data packet;
    queue = queue + 1;                                 /* put packet in the queue */
    sendPacket();
else                                                   /* else it must be check interval */
    sample the channel;
    if preamble is found ∧ (preamble is for me ∨ a broadcast one) then
        if upcoming data packet is redundant then
            send NACK;
        end
        else
            send pre-ACK;
            wait for a data packet;
            if data packet is received then
                queue = queue + 1;                     /* push packet into the queue */
                if moreToFollow == 1 then              /* packets to follow */
                    send pre-ACK;
                    receive data packet;
                    queue = queue + 1;
                end
                sendPacket();
            end
        end
    end
end
dutyCycle();                                           /* go back to sleep mode */
```

and know their locations. They are either deployed in an order or they know their physical locations with respect to few reference nodes. The selection and working of reference nodes is out of our scope. We also assume that the density of nodes is well enough that a node can communicate with multiple neighbors. All nodes carry unique node IDs. The sink node does not have any energy or storage restrictions, whereas other nodes have limited and non-replicate energy resources.

A node using AREA-MAC can play three different roles, i.e., a source node that generates and sends the packet, a destination node (mostly the sink node) that is the ultimate target of the packet, and a forwarding node that is neither a generator nor a destination node but is an intermediate node that forwards a packet to the next hop. Forwarding nodes are temporarily considered as sender/target nodes for the specific communication in which they are involved. These roles are drawn out in an understanding way in Figure 6.8.

Algorithm 6.2: sendPacket()

```
pc = 0;                                              /* preamble counter */
bc = burstCount;                                        /* burst counter */
moreToFollow = false;                           /* for adaptive duty cycling */
packetsToSend = queueSize;                    /* total packets in the queue */
if packetsToSend ≥ 1 then  /* packet must be there in the queue */
    perform carrier sense;
    if channel is clear then
        for pc ← 1 to bc do
            send a preamble;
            wait for pre-ACK;
            if got NACK then                        /* packet is redundant */
                queue = queue − 1;    /* pull packet from the queue */
                break;
            end
            if got pre-ACK then
                if packetsToSend ≥ 2 then
                    set moreToFollow; /* wants to send more packets */
                end
                send data packet;
                queue = queue − 1;
                if moreToFollow == true then
                    receive pre-ACK;
                    send data packet;
                    queue = queue − 1;
                end
                break;
            end
        end
    end
end
```

Algorithm 6.3: dutyCycle()

```
if queueSize > 0 then          /* if packets are there to be sent out */
    perform shortSleep;                         /* go for a short sleep */
end
else
    perform longSleep;                      /* else go for a full sleep */
end
wakeupProcess();                              /* wake up after sleep */
```

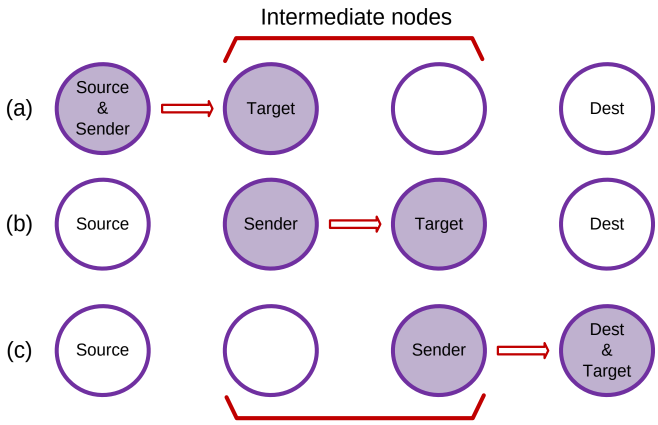


Figure 6.8: Different roles of an AREA-MAC node depending on when it generates, transmits, and receives a packet. (a) the source node generates and sends a packet to the first intermediate node, (b) the first intermediate forwards the packet to the second intermediate, and (c) the second intermediate forwards the packet to the destination node.

6.2.3 Network Model

An AREA-MAC based WSN can be represented by an undirected graph $G(V, E)$, called a connectivity graph, where $V = \{v_0, v_1, ..., v_{N-1}\}$ is the set of N sensor nodes and E is the set of links connecting nodes. Such a graph can be described as a grid topology of $m \times n$ order with m rows and n columns or as a random topology. Both grid and random topologies are drawn in Figure 6.9.

6.2.3.1 Grid Topology

For the grid topology, nodes are placed at the (x, y) location, where $1 \leq x \leq m$ and $1 \leq y \leq n$. If the grid location is given, the node ID can be determined by Equation (6.12). Alternatively, given the node ID, its location can be calculated by Equation (6.13).

$$ID(x, y) = (x - 1)n + (y - 1) \qquad 1 \leq x \leq m; 1 \leq y \leq n \tag{6.12}$$

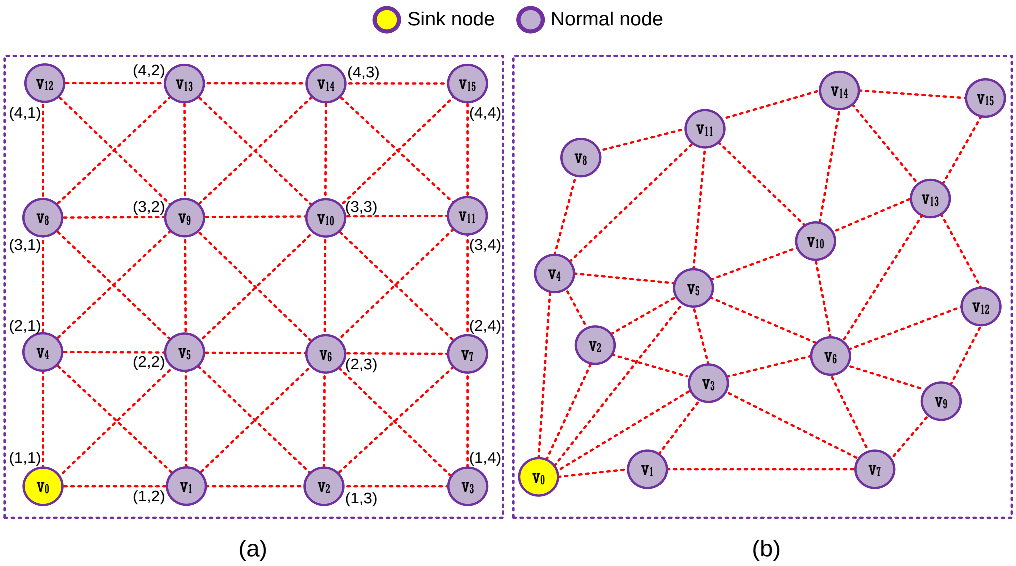


Figure 6.9: Portions of the (a) grid topology and (b) random topology for a WSN. All nodes are deployed in an ascending order with unique node IDs and know their locations (x,y). The deployment level decreases as the order or ID number of the node increases.

$$y = [ID\,(x,y) \bmod n] + 1$$
$$x = \frac{[ID\,(x,y) - y + 1]}{n+1} \tag{6.13}$$

The node v_0 represents the sink node, whereas nodes from v_1 to v_{N-1} represent the normal sensor nodes. An *up-level neighbor* of a node v_i is called a *1-level neighbor* (v_i^1) for v_i, if its location parameters (x, y) satisfy one of the following three conditions:

1. If $v_i^1(x) = v_i(x)$, then $v_i^1(y) = v_i(y-1)$
2. If $v_i^1(y) = v_i(y)$, then $v_i^1(x) = v_i(x-1)$
3. $v_i^1(x,y) = v_i(x-1, y-1)$

Similarly, an up-level neighbor of a node v_i is called a *2-level neighbor* (v_i^2) for v_i if its location parameters (x, y) satisfy one of the following three conditions:

1. $v_i^2(x) = v_i(x-2)$
2. $v_i^2(y) = v_i(y-2)$
3. $v_i^2(x,y) = v_i(x-2, y-2)$

The set $N^1_{v_i}$ contains all the 1-level neighbors and $N^2_{v_i}$ contains all the 2-level neighbors for the node v_i. For example, the 1-level and 2-level neighbors for the node v_{10} of the grid topology shown in Figure 6.9 are:

$$N^1_{v_{10}} = \{v_5, v_6, v_9\}$$
$$N^2_{v_{10}} = \{v_0, v_1, v_2, v_4, v_8\}$$

Since we assume a sufficient density of nodes, the node v_i can transmit packets to both of its up-level ($N^1_{v_i}$ and $N^2_{v_i}$) neighbors and can receive packets from its *low-level* neighbors. Low-level neighbors of the node v_i are the nodes for which v_i is either a 1-level or a 2-level neighbor. The degree of the node v_i, denoted by δ_{v_i}, represents the total number of up-level neighbors for the node and is equal to $|N^1_{v_i}| + |N^2_{v_i}|$, hence $\delta_{v_{10}} = 8$ for the node v_{10} in the above example. A bidirectional wireless link exists between v_i and its neighbor v_j and is represented by an edge $(v_i, v_j) \in E$. The euclidean distance between the node v_i and its neighbor v_j is given by ∂_{v_i,v_j}, and their connectivity is shown by the binary variable C_{v_i,v_j}.

6.2.3.2 Random Topology

Figure 6.9(b) represents a random topology for an AREA-MAC based WSN, where sensor nodes are randomly deployed. We assume that nodes are deployed in an ascending order with respect to their distances from the sink node, sometime it might be difficult to have such a deployment in reality though. The sink node (v_0) gets the highest deployment level (the deployment level decreases as the order or ID number of the node increases.). Sensor nodes learn about their location and up-level neighbors during the network setup phase. Hence, like the grid topology, nodes in the random topology have 1-level and 2-level neighbors. For example, the 1-level and 2-level neighbors for the node v_{10} in Figure 6.9(b) are:

$$N^1_{v_{10}} = \{v_3, v_5, v_6\}$$
$$N^2_{v_{10}} = \{v_0, v_1, v_2, v_4\}$$

The degree of the node v_{10} for the random topology is $\delta_{v_{10}} = 7$.

6.2.4 Traffic Model

Nodes using AREA-MAC generate data packets at the application layer by using two type of traffic generating scenarios, namely *simple traffic* and *burst traffic*. With the simple traffic scenario, a node generates a random data packet at every t seconds with the deviation of τ. For the evaluation of AREA-MAC, we choose $t = 75$ and $\tau = 45$, which means that for an experiment run of 1800 seconds, a node generates between 15 to 60 data packets for the simple traffic scenario. For the burst traffic scenario, the node generates double amount of data packets with the same rate and deviation, i.e., it generates from 30 to 120 data packets for the duration of 1800 seconds.

As stated, AREA-MAC classifies data traffic into *normal messages* and *real-time messages*. Nodes mostly generate normal messages, but occasionally they generate real-time messages. We calculate the inter-arrival time of real-time message on the basis of the Poisson distribution. If the average rate of data packet generation is λ, then the probability that there are exactly k real-time messages is given by:

$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad k \geq 0 \tag{6.14}$$

Where e is the base of the natural algorithm and k is the non-negative integer number representing occurrences of a real-time event. For $t = 75$, $\tau = 45$, and the experiment time of 1800 seconds, the value of λ ranges between 15 to 60. We consider that the maximum value of k is either equal to or less than 1/3 of λ.

6.2.5 Dissemination Model

As discussed in Section 2.4.3.4, the energy consumption of a node increases linearly with the increasing distance between the sender and receiver. By considering a simplistic example in Equation (2.1) through Equation (2.5), we have shown that a significant amount of energy can be saved by forcing packets to travel via multi-hop paths. However, after accumulating processing energy, collision probability, and other factors to this example, the amount of energy saving via multi-hop communication would not remain as significant as it is anticipated. We believe it would still be a positive value, therefore AREA-MAC adapts multi-hop communication among nodes, where they receive packets from their neighbors and forward to the sink node.

Additionally, AREA-MAC supports both broadcast and unicast form of communication. Under the broadcast dissemination, which we call a *Norm-pre* scenario, nodes simply broadcast preambles, and the first node that receives the preamble responds the sender and starts with the reception process. For the unicast dissemination, which is referred to as a *Dest-pre* scenario, preambles are sent to the specific destinations usually in the direction of the sink node, i.e., towards the up-level neighbors. For the Dest-pre scenario, the application can choose whether to forward data to 1-level or to 2-level neighbors. We analyze the effect of both broadcast and unicast disseminations in Chapter 7.

6.2.6 Energy Model

A node using AREA-MAC remains in sleep mode most of the time and wakes up shortly to sample the channel. We divide the system time T in small discrete and equidistant intervals, $t_0, t_1, ..., t_n$. The node wakes up at each check interval, I_{ci}, for a wake-up interval duration, T_{wk}, and samples the channel for a T_{sa} period.

If the node finds a preamble on the channel, it starts with the receiving process that lasts for a T_{rx} duration. If the node has a data packet to send out, it first senses the carrier for a T_{cs} period. On finding the carrier idle, the node starts with the transmitting process for a T_{tx} duration. Alternatively, it goes back to sleep mode for a T_{sl} duration, if the node does not find a preamble on the channel or it does not have any packet to transmit. The total energy consumption of the node per unit of time, E_t, is given by its energy consumption in channel sampling, carrier sensing, receiving packets, transmitting packets, sleeping, radio switching, and processing packets respectively, and is formulated in Equation (6.15). Note that all the notations are given in Table 6.1.

$$E_t = E_{sa} + E_{cs} + E_{rx} + E_{tx} + E_{sl} + E_{sw} + E_{pr} \tag{6.15}$$

$$E_t = P_{sa}T_{sa} + P_{cs}T_{cs} + P_{rx}T_{rx} + P_{tx}T_{tx} + P_{sl}T_{sl} + P_{sw}T_{sw} + P_{pr}T_{pr} \tag{6.16}$$

Equation (6.16) shows the power consumption and the time spent by the node in the respective state for a given time interval. Next, we calculate the individual time spent by the node in each of the state.

At every check interval duration, the node wakes up and performs the channel sampling. The time required to sample the channel is given by:

$$T_{sa} = \frac{1}{I_{ci}} T_{sa}^{av} \tag{6.17}$$

At each data generating interval, the node generates a data packet. Before sending out the packet, it performs the carrier sense to observe whether the channel is clear.

$$T_{cs} = \frac{1}{I_{dt}} T_{cs}^{av} \tag{6.18}$$

Where

$$T_{cs}^{av} = T_{bo} = S_{cw}T_{st} \tag{6.19}$$

The time spent in the receive mode by the node is equal to the sum of the times the node spends in receiving data packets, preambles, and pre-ACK frames and is given by Equation (6.20). Note that the node can receive packets during a normal or an adaptive process and can receive an overheard packet either it be a preamble, data, or a pre-ACK frame.

$$T_{rx} = T_{rx}^{pr} + T_{rx}^{dt} + T_{rx}^{ak} \tag{6.20}$$

where

$$T_{pr}^{rx} = L_{pr}T_{bt} \tag{6.21}$$

$$T_{dt}^{rx} = L_{dt} T_{bt} \tag{6.22}$$

$$T_{ak}^{rx} = L_{ak} T_{bt} \tag{6.23}$$

The time spent in the transmit mode for the node is the sum of the times spent in sending data packets, preambles, and pre-ACK frames and is given in Equation (6.24). Like the receiving time, the node can transmits packets via a normal or an adaptive process.

$$T_{tx} = T_{tx}^{pr} + T_{tx}^{dt} + T_{tx}^{ak} \tag{6.24}$$

Where

$$T_{tx}^{pr} = L_{pr} T_{bt} \tag{6.25}$$

$$T_{tx}^{dt} = L_{dt} T_{bt} \tag{6.26}$$

$$T_{ak}^{tx} = L_{ak} T_{bt} \tag{6.27}$$

E_{pr} in Equation (6.15) represents the radio processing energy of the node. This energy does not change with the radiated power, unlike E_{tx} that changes with the output power. The radio needs E_{pr} energy for the working of oscillators, mixers, filters, other components, and this value is highly platform and implementation specific and may vary from ηJ/bit to μJ/bit [125] (η for nano and μ for micro). We calculate the radio processing energy of a node depending on the total packets it has processed (received and transmitted). If the node receives and transmits a total number of n_1 preambles, n_2 data packets, and n_3 ACK frames during a time unit, then its processing time can be given as:

$$T_{pr} = (n_1 L_{pr} + n_2 L_{dt} + n_3 L_{ak})\, T_{pr}^{bt} \tag{6.28}$$

The node goes to the sleep mode, if it is not doing anything else.

$$T_{sl} = 1 - (T_{sa} + T_{cs} + T_{rx} + T_{tx} + T_{sw} + T_{pr}) \tag{6.29}$$

6.2.6.1 System Lifetime

In Chapter 4, we have defined several possible definitions of the system lifetime for a WSN. However, the selection of a proper system lifetime criterion is application-specific, which strongly depends on the lifetimes of the individual nodes. Therefore, in order to increase the system lifetime of a WSN, we simply reduce the energy consumption at each node (except the sink node).

$$\min_{v_i \in V} E_t^{v_i} \qquad i > 0 \tag{6.30}$$

6.2.7 Delay Model

After calculating the energy consumption of a node, now we calculate the end-to-end delay for each data packet that the node generates. The end-to-end delay of a packet is defined as the time when the source node v_i generates and puts the packet in the queue to the time the sink node v_0 receives it. Since AREA-MAC nodes transmit packets via multi-hop paths, the end-to-end packet delay can be further divided into three parts, namely delay at the source node, delay at the intermediate nodes, and delay at the destination node.

$$D_{v_i,v_0} = D_{v_i} + D_{fd} + D_{v_0} \qquad i > 0 \tag{6.31}$$

The delay in sending out a data packet at the source node v_i is given by:

$$D_{v_i} = \begin{cases} T_{tx}^{nr} + T_{qu}, & \text{for normal processing} \\ T_{tx}^{ad} + T_{qu}, & \text{for adaptive process} \end{cases} \tag{6.32}$$

Where T_{qu} is the queuing delay that represents the time for which a data packet is kept in the queue before it gets a chance to be transmitted. The queuing delay mainly depends on the traffic load of the node. The delay at the forwarding nodes, D_{fd}, is given by the sum of delays at all the forwarding nodes. Let F be the set containing all the forwarding nodes, each denoted by v_f, such that $v_f \in F \subset V$, then D_{fd} is given by:

$$D_{fd} = \begin{cases} T_{rx}^{nr} + T_{fd}^{nr} + T_{qu}, & \text{for each } v_f\text{; for normal processing} \\ T_{rx}^{ad} + T_{fd}^{ad} + T_{qu}, & \text{for each } v_f\text{; for adaptive processing} \end{cases} \tag{6.33}$$

Finally, the delay at the destination (sink) node v_0 is given by:

$$D_{v_0} = T_{rx} + (L_{pr} + L_{ak} + L_{dt})\, T_{pr}^{bt} \tag{6.34}$$

Since the sink node is always ready to receive a packet (by keeping the radio in the constant receive mode), it does not need to sample the channel.

6.3 Analytical Comparison of AREA-MAC with the Long Preamble Technique

Since AREA-MAC uses the short preamble method, it is more efficient in terms of energy consumption and packet delay as compared to the long preamble scheme. We next analytically study the energy and delay gains of the short preamble scheme over the long preamble scheme.

6.3.1 Energy Gain

For better understanding, we split energy gain into two parts: *transmit energy gain* and *receive energy gain*.

6.3.1.1 Transmit Energy Gain

With the long preamble technique, a sender first senses the carrier before initiating any transmission. On finding the carrier idle, it sends out an extended preamble lasting for a complete check interval duration, irrespective to the wake-up time of the intended receiver. After sending the preamble, the sender transmits a data packet and then switches its radio to the RX mode in order to receive the acknowledgment from the receiver. The radio transmit energy of the node in sending out a data packet for the long preamble technique, E_{tx}^{lp}, is given by:

$$E_{tx}^{lp} = T_{cs}P_{cs} + \left(T_{ci} + T_{tx}^{dt}\right) P_{tx} + T_{sw}P_{sw} + T_{rx}^{ak}P_{rx} \tag{6.35}$$

On contrary, the sender sends out a burst of short preambles rather than one long preamble with the short preamble scheme. It switches the radio to the RX mode after sending each preamble in order to receive the pre-ACK from the intended receiver. The sender transmits short preambles for the whole check interval duration, but stops sending further preambles as soon as it receives the pre-ACK. If we assume the transmission time in sending a short preamble is equal to the radio switching and waiting time in the RX mode, then the time until a sending node receives the pre-ACK can be equally divided into the preamble sending time and ACK waiting time. However, for simplicity we integrate the radio switching and waiting time into the preamble transmission time. Hence, the radio transmit energy in sending out a data packet for the short preamble technique, E_{tx}^{sp}, can be given as:

$$E_{tx}^{sp} = T_{cs}P_{cs} + \left(nT_{tx}^{pr} + T_{tx}^{dt}\right) P_{tx} + T_{rx}^{ak}P_{rx} \tag{6.36}$$

Since AREA-MAC is an asynchronous protocol where both sender and receiver are left unsynchronized, a sender can receive the pre-ACK at anytime during the check interval duration. It could be at the beginning of the check interval (the best case), at the half of the check interval (the average case), or at the end of the check interval duration (the worst case). If we assume four different preamble sending levels, such that p_a, p_b, p_c, and p_d denote the probabilities that the sender receives the pre-ACK frame at T_{ci}, $T_{ci}/2$, $T_{ci}/4$, and $T_{ci}/8$ time respectively, then Equation 6.36 can be written as:

$$E_{tx}^{sp} = T_{cs}P_{cs} + \left[T_{ci}\left(p_a + \frac{p_b}{2} + \frac{p_c}{4} + \frac{p_d}{8}\right) + T_{tx}^{dt}\right] P_{tx} + T_{rx}^{ak}P_{rx} \tag{6.37}$$

$$E_{tx}^{sp} = T_{cs}P_{cs} + \left[\frac{T_{ci}\left(8p_a + 4p_b + 2p_c + p_d\right)}{8} + T_{tx}^{dt}\right] P_{tx} + T_{rx}^{ak}P_{rx} \tag{6.38}$$

If

$$p_a + p_b + p_c + p_d = 1 \tag{6.39}$$

Then

$$8\,(p_a + p_b + p_c + p_d) = 8 \tag{6.40}$$

which implies that

$$8p_a + 4p_b + 2p_c + p_d < 8 \tag{6.41}$$

$$\frac{8p_a + 4p_b + 2p_c + p_d}{8} < 1 \tag{6.42}$$

$$\Rightarrow \quad E_{tx}^{sp} < E_{tx}^{lp} \tag{6.43}$$

Hence, the radio of an AREA-MAC node consumes less time in the transmit mode as compared to the traditional channel polling scheme.

6.3.1.2 Receive Energy Gain

A receiving node working under the long preamble technique wakes up at each check interval duration to sense the channel. Once it senses the preamble, it stays awake until the data packet arrives. The receiving node can receive the preamble at any time during the check interval duration of the sending node. On average, the receiver acquires a preamble at the half of the check interval duration of the sending node. Hence, the energy consumption of the radio receiver for the long preamble technique while receiving a data packet, E_{rx}^{lp}, is given as:

$$E_{rx}^{lp} = T_{sa}P_{sa} + \left(\frac{T_{ci}}{2} + T_{rx}^{dt}\right) P_{rx} + T_{sw}P_{sw} + T_{tx}^{ak}P_{tx} \tag{6.44}$$

On the other side, the receiving node working under the short preamble technique also wakes up at a check interval duration. It transmits the pre-ACK frame to the sender as soon as it receives the short preamble from the sender. After sending the pre-ACK, the receiver switches its radio to the RX mode to receive the data frame. The energy consumption of the radio receiver for the short preamble technique while receiving a data packet, E_{rx}^{sp}, is given as:

$$E_{rx}^{sp} = T_{sa}P_{sa} + \left(T_{rx}^{pr} + T_{rx}^{dt}\right) P_{rx} + T_{sw}P_{sw} + T_{tx}^{ak}P_{tx} \tag{6.45}$$

A node can receive a short preamble at any time during the preamble transmission. Unlike the long preamble technique, the time the node receives the preamble does not matter a lot for the short preamble scheme, as the receiver either acknowledges the sender via a pre-ACK as soon as it receives the short preamble or switches back to the pre-sleep mode, if the preamble is not destined to it. Therefore, an AREA-MAC node in any case receives only one short preamble. Moreover, the energy consumption at the receiving node is totally dissociated from the duty cycle value of nodes. It is unlike the long preamble scheme, where the size of the preamble totally depends on the duty cycle value, i.e., on the check interval duration of the node. For example, with the duty cycle of 1000 ms, the length of a long preamble must be at least 1000 ms, whereas the size of a short preamble remains fixed, i.e., 14 byte long in our case. If the receiving node wakes up on average half of the check interval duration, it must remain awake for other 500 ms for the long preamble technique before it receives (or rejects) a data packet. Therefore, we can easily assume that a long preamble is convincingly several times longer than a short preamble and conclude that:

$$T_{rx}^{pr} < \frac{T_{ci}}{2} < T_{ci} \tag{6.46}$$

$$\Rightarrow \quad E_{rx}^{sp} < E_{rx}^{lp} \tag{6.47}$$

6.3.2 Delay Gain

The average per-hop packet delay of the long preamble technique at the transmitting node, D_{av}^{lp}, is given as:

$$D_{av}^{lp} = T_{qu} + T_{cs} + T_{ci} + T_{tx}^{dt} + T_{rx}^{ak} \tag{6.48}$$

Note that the sending node transmits the preamble for a complete check interval duration. Alternatively, with the short preamble technique, the sender can receive a pre-ACK and then transmits a data packet to the intended receiver at any time during the check interval duration, i.e., after transmitting n short preambles. Hence, the average per-hop delay of the short preamble technique, D_{av}^{sp}, is given by:

$$D_{av}^{sp} = T_{qu} + T_{cs} + nT_{pr}^{tx} + T_{tx}^{dt} + T_{rx}^{ak} \tag{6.49}$$

Considering the method used in Section 6.3.1.1, if we assume p_a, p_b, p_c and p_d the probabilities that the receiver receives the short preamble at T_{ci}, $T_{ci}/2$, $T_{ci}/4$ and $T_{ci}/8$ time respectively, then

$$D_{av}^{sp} = T_{qu} + T_{cs} + T_{ci}\left(p_a + \frac{p_b}{2} + \frac{p_c}{4} + \frac{p_d}{8}\right) + T_{tx}^{dt} + T_{rx}^{ak} \tag{6.50}$$

$$D_{av}^{sp} = T_{qu} + T_{cs} + T_{ci}\frac{(8p_a + 4p_b + 2p_c + p_d)}{8} + T_{tx}^{dt} + T_{rx}^{ak} \tag{6.51}$$

If

$$p_a + p_b + p_c + p_d = 1 \tag{6.52}$$

then

$$8\,(p_a + p_b + p_c + p_d) = 8 \tag{6.53}$$

which implies that

$$8p_a + 4p_b + 2p_c + p_d < 8 \tag{6.54}$$

$$\frac{8p_a + 4p_b + 2p_c + p_d}{8} < 1 \tag{6.55}$$

$$\Rightarrow \quad D_{av}^{sp} < D_{av}^{lp} \tag{6.56}$$

Hence, per-hop packet delay for AREA-MAC nodes is less than the long preamble technique. This difference gets better as the number of hop increases and as the probabilities p_d and p_c increase. Furthermore, the adaptive duty cycle scheme of AREA-MAC, where the sending node transmits a data packet even without the need of carrier sense and preamble transmission, significantly reduces packet delay for nodes. The per-hop delay for the adaptive duty cycle of AREA-MAC, D_{ad}^{sp}, is given by:

$$D_{ad}^{sp} = T_{qu} + T_{rx}^{ak} + T_{tx}^{dt} \tag{6.57}$$

Prioritizing a message with AREA-MAC further improves delay of the real-time messages.

6.3.3 Energy and Delay Gain Presentation

Figure 6.10 depicts AREA-MAC gain in terms of energy and time over the protocols working on the long preamble technique. The conventional preamble sampling scheme has been compared with both normal and adaptive processing of AREA-MAC nodes. The figure elucidates that the AREA-MAC gain is not only limited to the sender and receiver, non-target receivers also substantially benefit from it. For the adaptive processing, where the sender directly transmits a data packet to the receiving node without transmitting a single preamble, the gain is significantly eminent for the sender, receiver, and non-target nodes.

By applying Equation (6.35) through Equation (6.57) on the parameter setting shown in Table 6.2, we figure out the energy and delay differences between the short and long preamble schemes. Figure 6.11 compares the transmission time and per-hop delay in sending out a data packet for both schemes against different check interval values. For the short preamble scheme, all the four discussed probabilities, i.e., p_a, p_b, p_c, and p_d have been considered. As expected the probability p_a,

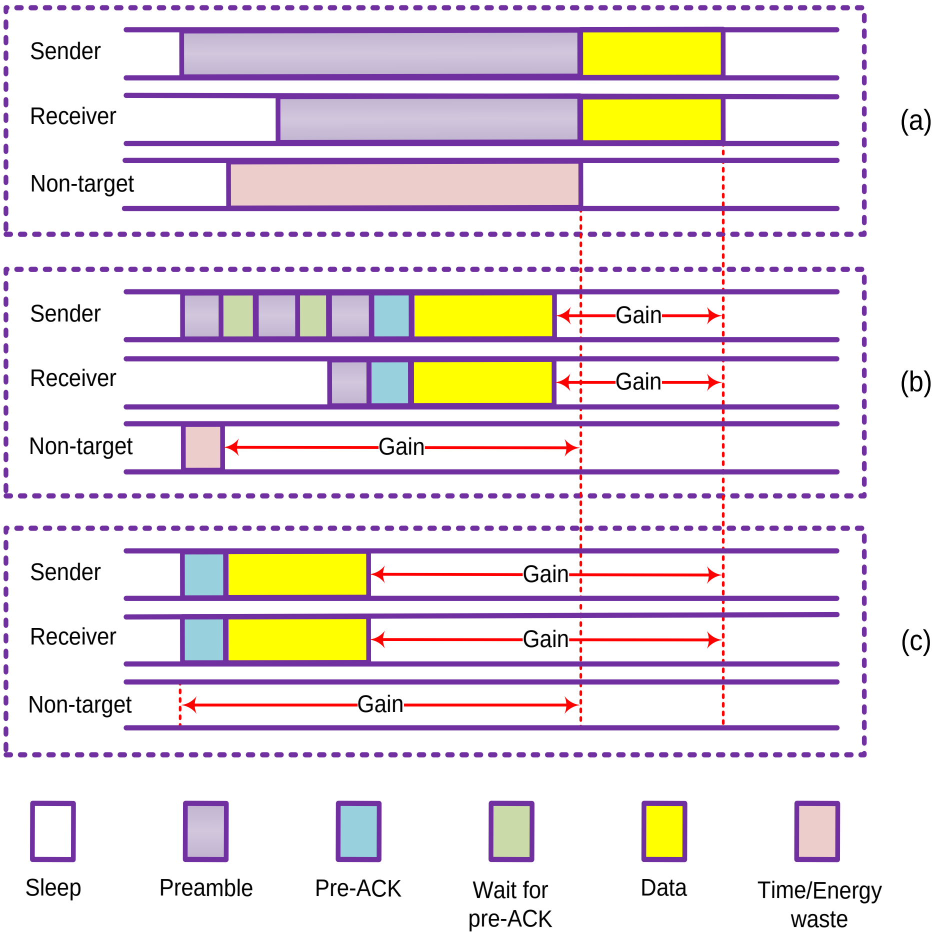


Figure 6.10: A substantial time/energy gain of AREA-MAC over the long preamble technique. The figure shows the transmission and reception processes at the sender, receiver, and non-targeted receivers for (a) the long preamble technique, (b) for AREA-MAC with the normal processing, and (c) for AREA-MAC with the adaptive processing.

Table 6.2: Parameters setting for the analytical evaluation

Parameter	**Value**
Short Preamble	14 bytes
Data Packet	42 bytes
ACK Frame	12 bytes
Carrier Sense Time	2 ms
Channel Sampling Time	1 ms
Radio Switching Time	0.3 ms
Byte Transmit Time	0.032 ms
Check Interval	Varying

which represents the worst case scenario for the short preamble scheme, has almost the same energy consumption and per-hop packet delay as of the long preamble scheme. However, p_b, p_c, and p_d sequentially improve both the transmission energy and packet delay for sensor nodes. Figure 6.11 also manifests that the radio transmission time and per-hop packet delay for both schemes are proportional to the check interval time of a node. However, due to the transmission of extended preambles, the long preamble scheme results in at least the check interval size transmit time and per-hop packet delay, which have been considerably decreased with the short preamble scheme.

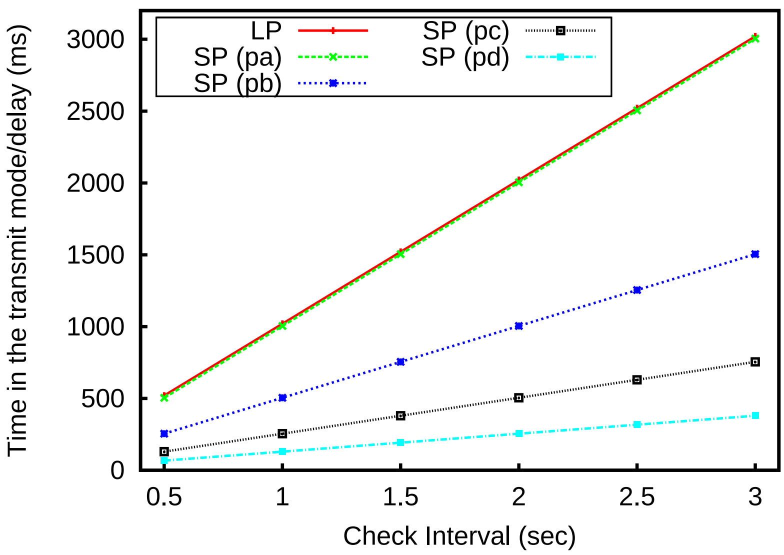


Figure 6.11: Energy (in terms of radio transmit time) and delay gains of AREA-MAC using the short preambles as compared to the long preamble scheme for different check interval values. pa, pb, pc, and pd are the preamble receiving probabilities of the short preamble scheme. Note that SP stands for the short preamble and LP for the long preamble scheme and both LP and SP with pa are almost having identical values in this graph. On average, AREA-MAC protocol achieves lower radio transmit time and packet delay for nodes than the long preamble scheme.

Figure 6.12 shows the time the radio of a node spends in the receive mode in order to receive a data packet for both long and short preamble schemes. It is obvious from the figure that the radio receiving time of both targeted receivers and non-targeted nodes is considerably reduced and is dissociated from the duty cycle value of the node for AREA-MAC protocol. On contrary, nodes with the long preamble scheme, whether they are intended receivers or non-targeted nodes, squander substantially higher time in receiving long preambles, which exponentially increases with the increasing duty cycle value of nodes. We assume that nodes (targeted or non-targeted) working under the long preamble scheme wake up on average at a half of the check interval duration of the sender. Therefore, there is a possibility that the radio receiving time may get better (or alternatively even worsen) depending on the time the node senses the preamble. However, the preamble sensing time does not really matter for the energy consumption of both targeted and non-targeted AREA-MAC nodes, as they receive and process only one short preamble.

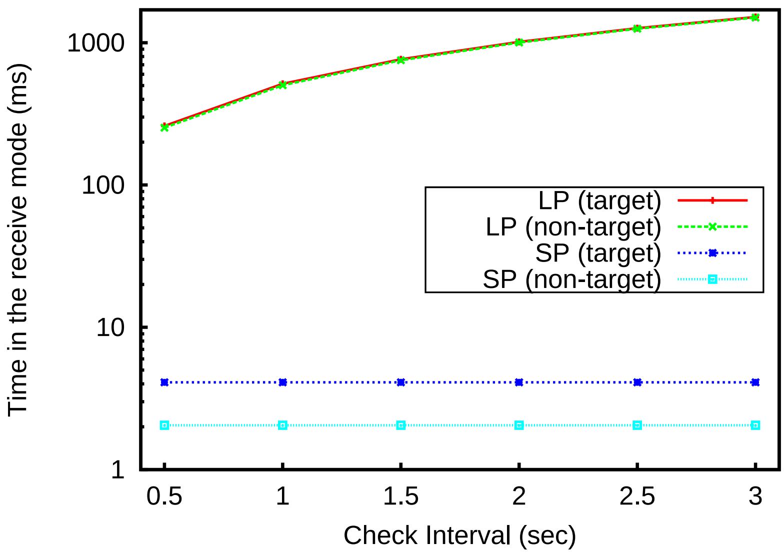


Figure 6.12: Energy (in terms of the radio receive time) gain of AREA-MAC against the long preamble technique for different check interval values for both targeted and non-targeted nodes. Note that SP stands for the short preamble and LP for the long preamble scheme and both LP (target) and LP (not-target) are almost having identical values in this graph. Hence, AREA-MAC protocol working under the short preamble scheme significantly reduces the radio receive time of targeted as well as non-targeted nodes.

Figure 6.13 depicts the duty cycle value of the long preamble scheme and of four considered probabilities of the short preamble scheme for an assumed system time of 1800 seconds. We consider different traffic conditions for this case, where each node receives and transmits 10, 20, 30, 40, 50, and 60 data packets respectively from and to a neighbor. Note that packet forwarding, packet overhearing, and processing energy of nodes is not considered here. AREA-MAC achieves lower

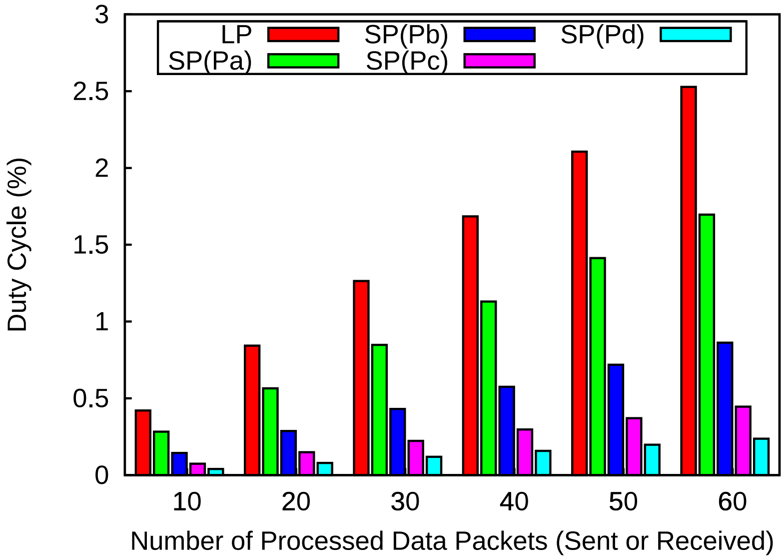


Figure 6.13: Duty cycle gain of the considered probabilities of AREA-MAC protocol against the long preamble scheme for different traffic conditions. As anticipated, AREA-MAC achieves much lower duty cycle for nodes as compared to the long preamble scheme.

duty cycle for a given number of data packets as nodes remain in the sleep mode for a longer period than the long preamble scheme. The duty cycle of nodes gets better with the increasing probabilities of p_c or p_d. Moreover, if nodes keep forwarding a packet until it is received by the sink node, then the long preamble scheme consumes a lot of radio time in transmit and/or receive modes, especially with broadcast communication and ultimately achieves much higher duty cycle values than the short preamble scheme.

6.4 Conclusion

The polling based channel accessing technique - even though it saves significant energy as compared to other channel accessing techniques - incurs several limitations in terms of energy and delay inefficiencies, mainly causing from its long and extended preambles. This triggers the need to cut down the length of long preambles. In this chapter, we have proposed AREA-MAC protocol for WSNs that supports the transmission of short preambles in order to negate the drawbacks of the traditional channel polling scheme. AREA-MAC is a distributed MAC protocol for WSNs that achieves significant gains in terms of energy and delay for sensor nodes by substituting long preambles with the short ones and by putting extra information with each preamble. The protocol attains reduced energy consumption in several ways that include reduction in idle listening, overhearing, collisions, over-emitting, and control overhead. AREA-MAC results in improved packet delay for sensor nodes by offering several features such as pre-ACK, pre-sleep, adaptive duty cycling, short and long sleep modes, message prioritizing, and convergecast communication. Moreover, AREA-MAC reduces packet redundancy, increases packet delivery ratio, supports suitable adaptability and scalability, and offers different routing and traffic generating schemes for a WSN.

In the coming chapters, the implementation and performance evaluation of AREA-MAC protocol on a simulation as well as the real testbed platform are elaborated.

Chapter 7

Performance Evaluation of AREA-MAC: A Simulation Approach

A good simulation, be it a religious myth or scientific theory, gives us a sense of mastery over experience.

–Heinz Rudolf Pagels

This chapter presents the simulation implementation and performance evaluation of AREA-MAC protocol by first explaining the simulation model and configuration, and then discussing various evaluation results related to several metrics. These metrics include energy consumption, packet delay, packet delivery, preamble transmission and reception, and overheard packets for sensor nodes. Moreover, effects of varying traffic, topology, scaling, routing, and other factors on the performance of AREA-MAC protocol have also been elaborated. A comparison between AREA-MAC and B-MAC protocols has been presented. Since B-MAC belongs to the same family of AREA-MAC protocol (both use the channel polling scheme), it makes the comparison between them meaningful.

7.1 Simulation Setup

As mentioned in Section 4.1.2, we have implemented AREA-MAC on the OMNeT++ simulation platform [48] with the help of the Mobility Framework [103]. This section describes the setup of the simulation system by explaining the simulation model, simulation environment, and parameters selection.

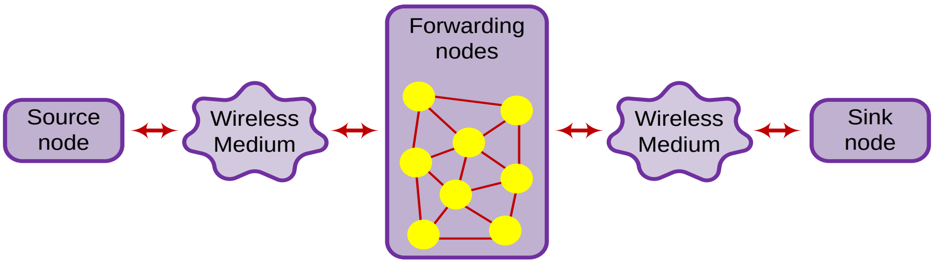


Figure 7.1: An overview of the AREA-MAC simulation framework, where the source node generates data packet and sends it to a neighbor node over the wireless channel. The neighbor forwards the packet further towards the sink node.

7.1.1 Simulation Model

Figure 7.1 shows the big picture of the AREA-MAC simulation implementation, where a source node generates a data packet and transmits it over the wireless medium. A neighboring node, which is in the range of the source node, receives the packet and forwards it to the sink node.

The sink node is the ultimate destination of all data packets generated by each node. Since the sink node does not have any energy limitations, it remains in receive mode throughout a simulation run (except when it sends an acknowledgement to the source node). Hence, the sink node does not send any preamble or a data packet, it only receives packets from other nodes. Understandably, the sink node receives a higher number of preambles and data packets as compared to other nodes, which perform duty cycle and switch their radios between wake-up and sleep modes.

Figure 7.2 presents the detailed simulation framework for the AREA-MAC implementation based on the Mobility Framework. This framework comprises of various modules, each one is briefed below.

- The *Environment Module* is related to the PHY layer parameters that include radio propagation, noise threshold, modulation scheme, carrier frequency, data rate, and transmission power of nodes. All these parameters are critical as their values define how close a simulation is to the real world implementation. We have used the probabilistic based Nakagami propagation model [107–109] for the simulation because probabilistic models achieve better and realistic results compared to deterministic models. The Nakagami propagation model, also called the Nakagami-m distribution, is a generic model that simulates certain environmental conditions in a better way to characterize signals sent over multipath fading channels [109, 126]. The probability distribution function of Nakagami model that draws the distribution of the received envelop x is given by:

$$P(x) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m x^{2m-1} e^{\left(-\frac{m}{\Omega}x^2\right)} \qquad x \geq 0; m \geq 1/2 \tag{7.1}$$

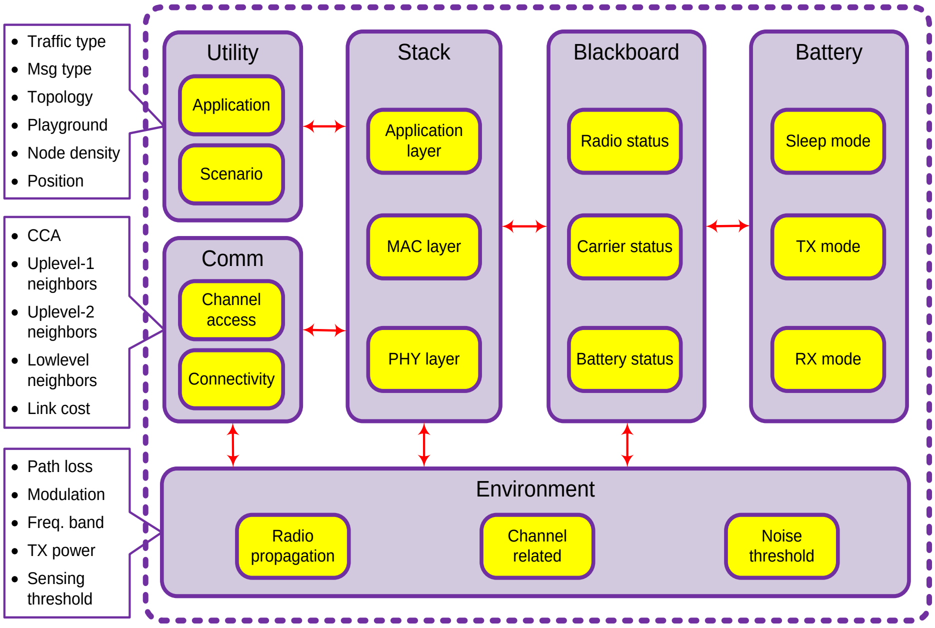


Figure 7.2: A detailed simulation framework of AREA-MAC protocol. Different modules communicate and cooperate with each other in order to achieve higher performance in terms of energy, latency, packet delivery, and other metrics.

Where $\Gamma(.)$ is the Gamma function. The parameter Ω, which is a scaling parameter and controls the spread of the distribution, reflects the average received power and is given by:

$$\Omega = E\left[X^2\right] \tag{7.2}$$

The parameter m, which is called the fading or shape factor, represents the asperity of fading, i.e., it is a measure of channel quality. The value of m is given by:

$$m = \frac{\Omega^2}{E\left[(X^2 - \Omega)^2\right]} \tag{7.3}$$

- The *Stack Module* reflects the working of different layers of the communication protocol stack, which is shown in Figure 2.4. Being a MAC protocol, the main functionalities of AREA-MAC reside on the MAC layer. However, nodes generate data traffic at the application layer, and like any other MAC protocol, AREA-MAC has a strong coupling with the PHY layer. To efficiently handle the working of the radio, AREA-MAC periodically interacts with the PHY layer in order to distinguish between different radio and carrier states.

- The *Utility Module* helps a node in deciding the type of data traffic (simple or burst), the type of the message (normal or real-time), topology (grid or random), node location, and the size of the playground.
- The *Communication Module* helps a node in contending with other nodes to gain access of the channel, in selecting connectivity between neighbors, and in distinguishing between up-level and low-level neighbors.
- The *Blackboard Module* provides a cross-layering functionality by offering an information exchange among different layers. This module works on the publish/subscribe pattern, where a node subscribes for an interested parameter (for example, the current status of the radio, channel, or battery), and the Blackboard Module publishes the related information to the node.
- The *Battery Module* updates power related statistics of a node every time the radio switches between the transmit, receive, or sleep mode. It uses three parameters; *sleepPower*, *receivePower*, and *sendPower*, which reflect the needed power (mA/sec) for the respective radio mode. The current level of the battery is published to the node via the Blackboard Module.

7.1.2 Simulation Configuration

AREA-MAC nodes use three types of frames to communicate with each other, namely preamble, data, and pre-ACK frames. Figure 7.3 represents the bit configuration of each of the frames. Note that the size of each frame is chosen in

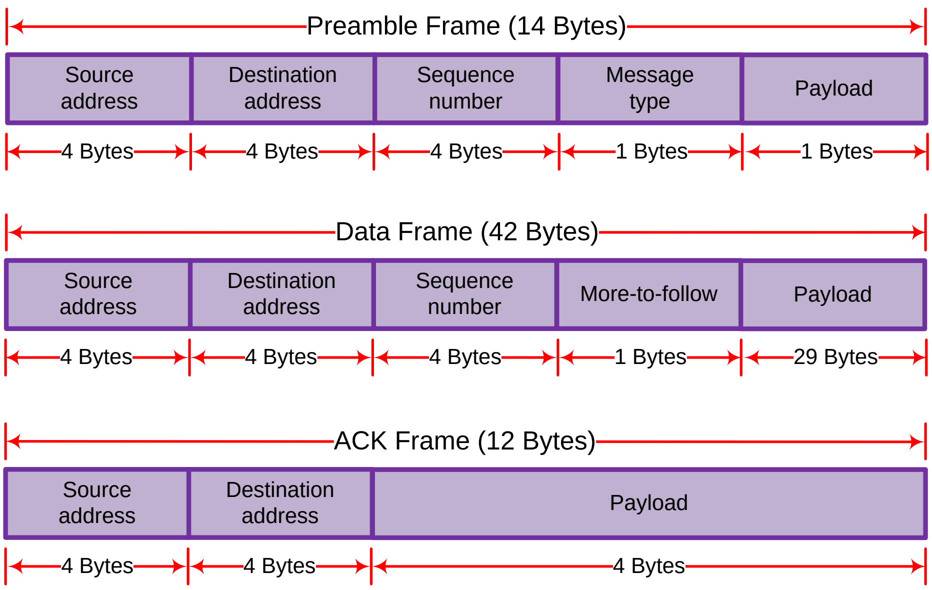


Figure 7.3: Format of preamble, data, and ACK frames used for the AREA-MAC implementation on the OMNeT++ simulation platform.

relevance with the testbed implementation of AREA-MAC protocol (Chapter 9). Table 7.1 points out values of various simulation parameters that have been used for the respective layer of the AREA-MAC implementation. Since AREA-MAC has been evaluated for a diverse number of network scenarios, values of some of the parameters vary as per the scenario requirements.

Table 7.1: Simulation configuration for the AREA-MAC implementation on the OMNeT++ platform.

Layer	**Parameter**	**Value**
App layer	Playground size	800x800, 1200x1200 m^2
	Number of nodes	16, 36, 64
	Topology	Grid, Random
	No. of replications	11
	Simulation time/run	1800 s
	Data generating interval	75, 25 s
	Data generating deviation	45, 15 s
	Data packet length	42 bytes
	Data burst size	1, 2
	RNG algorithm	Mersenne Twister
	Number of RNG streams	4
MAC layer	Queue size	10
	Check interval duration	1 sec
	Preamble length	14 bytes
	ACK length	12 bytes
	CCA threshold	-90 dBm
	Long sleep	Check interval duration
	Short sleep	1 ms
PHY layer	Transmission power	1 mW (0 dBm)
	Carrier frequency	868 MHz
	Data rate	400 Kbps
	Signal attenuation threshold	-110 dBm
	Propagation model	Nakagami

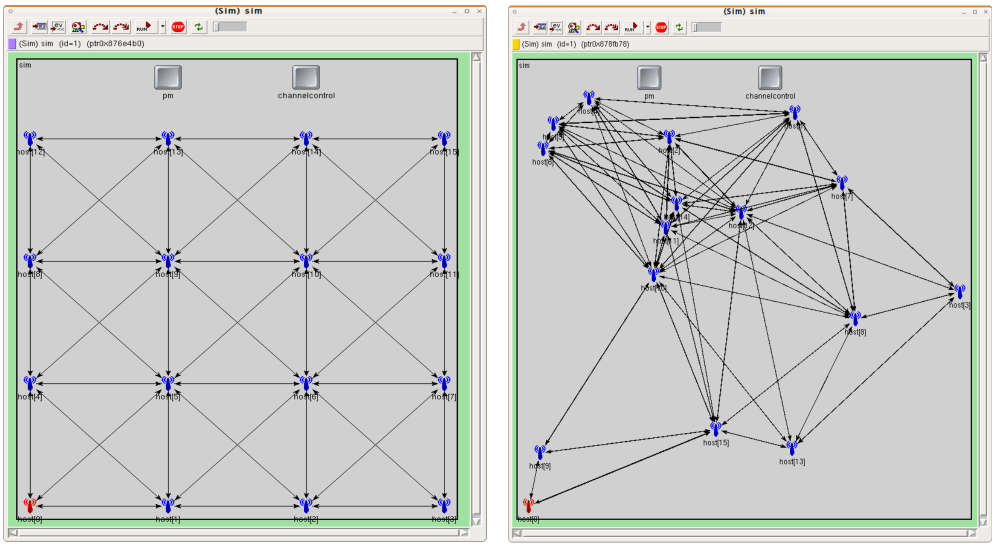
7.2 Evaluation Results

This section presents various simulation results to evaluate AREA-MAC protocol. Effects of topology changes, scalability, cross-layering (with different traffic and routing scenarios), and other factors have been analyzed. AREA-MAC protocol has also been compared with B-MAC protocol, which works on the long preamble scheme. As aforementioned in Chapter 6, two topologies, i.e., grid and random are used throughout the evaluation.

Unlike AREA-MAC protocol, where nodes broadcast preambles but send data packets to a specific node, B-MAC protocol broadcasts preambles as well as data packets. Hence B-MAC nodes receive more data packets than AREA-MAC nodes. Each received packet needs to be buffered into the receive buffer before gets processed. In order to efficiently deal with the duplicate/redundant packets, we set the queue size (receive buffer size) of B-MAC protocol as 20, whereas this value is equal to 10 for AREA-MAC protocol. The evaluated metrics for both protocols and topologies are the energy consumption, end-to-end delay, data packets delivery, preamble transmission and reception, and overheard packets. The considered scenarios are the normal traffic scenario, burst traffic scenario, high traffic scenario, varying routing scenario, medium size scaling scenario containing 36 nodes, and extended scaling scenario containing 64 nodes.

7.2.1 Normal Traffic Scenario

The normal traffic scenario is a simple and small size evaluating environment, where 16 nodes are deployed in a 800x800 m^2 area. For this scenario, both grid and random topologies of AREA-MAC protocol and a random topology of B-MAC protocol are considered. Figure 7.4 shows snapshots of both topologies for the OMNeT++ based AREA-MAC implementation. For the random topology, we keep the location of the sink node more or less fixed, whereas positions of other nodes have been randomly changed at the start of each experimentation run. The same is also true for the B-MAC random topology.



(a) Grid Topology (b) Random Topology

Figure 7.4: Snapshots of the OMNeT++ based (a) grid and (b) random topologies for the AREA-MAC implementation. The red-colored node at the bottom left corner represents the sink node, whereas normal nodes are the blue-colored one. Black lines show connectivity among nodes.

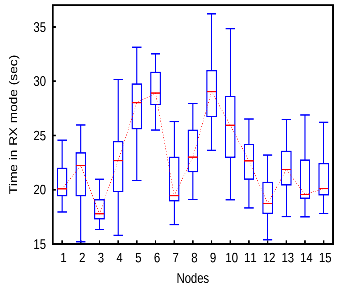
Each node except the sink node generates a single data packet (i.e., with the burst value of one) at every data generating interval with the given deviation value. The check interval duration for nodes is 1 second. This scenario is a simple broadcast scenario (the N_0 routing scheme), where nodes broadcast preambles prior to data packet transmission, and any neighboring node that is in the transmission range of the sending node may receive the packet irrespective to its location (or irrespective to the location of the sending node). We below present several evaluating results for all three considered network variants for the mentioned metrics.

7.2.1.1 Energy Consumption

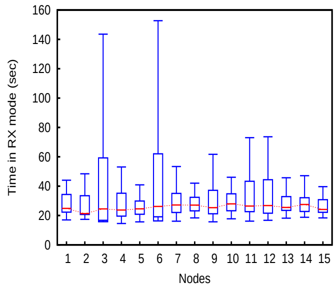
As discussed in Section 4.2.1, we calculate energy consumption of a node by counting the time its radio spends in receive, transmit, and sleep mode respectively. The carrier sense time of a node is included in the radio receive time. The higher the time the radio of a node is in sleep mode, the more is energy gain, and the better is its duty cycling. Since the sink node remains in constant receive mode, we do not include it in the energy related results that are depicted in Figure 7.5.

Figures 7.5a, 7.5b, and 7.5c show the radio receive time of nodes for the AREA-MAC grid topology, AREA-MAC random topology, and B-MAC random topology respectively. The radio receive time for B-MAC nodes is significantly higher (more than 30 times) than AREA-MAC nodes working under either of the topology. The two obvious reasons behind this distinction are: 1). B-MAC nodes, after sensing the existence of a preamble, stay in receive mode to receive the complete long preamble, and 2). B-MAC nodes broadcast preambles as well as data packets. Hence nodes receive more data packets as compared to AREA-MAC protocol, where nodes broadcast preambles but send data packets only to the specific nodes. For the AREA-MAC grid topology, nodes that are near the sink node, i.e., nodes with ID ranges between 1-2, 4-6, and 8-10 receive more preambles and data packets, particularly from the border nodes, and therefore they spend a slightly higher time in receive mode. Border nodes, i.e., nodes with IDs 3, 7, and 11-15 receive relatively less preambles and data packets, and hence they deplete less energy in receive mode. Nodes under the AREA-MAC random topology use up almost the same amount of time in receive mode; the slight difference is, however, due to their physical location.

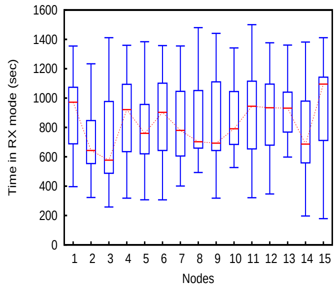
Since border nodes of the AREA-MAC grid are multi-hop away from the sink node and need to transmit more preambles until their respective neighbors wake up and acknowledge, their radios spend almost double time in transmit mode (Figure 7.5d) than that of the other grid nodes. Alternatively, nodes that are at the single-hop distance from the sink node understandably consume less time in transmit mode. Like the radio receive time, the radio transmit time of nodes working under the AREA-MAC random topology also remain, on average, same in Figure 7.5e. This time is, however, higher than that of the grid nodes and mainly depends on the distance of the node from the sink node. On contrary, B-MAC nodes deplete extremely higher radio time (more than 50 times of the



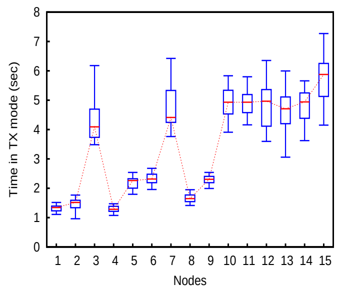
(a) Grid: Radio RX Time



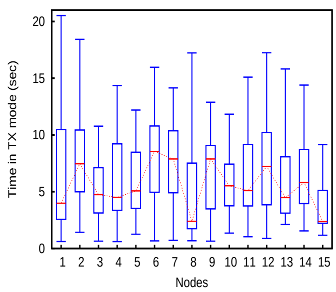
(b) Random: Radio RX Time



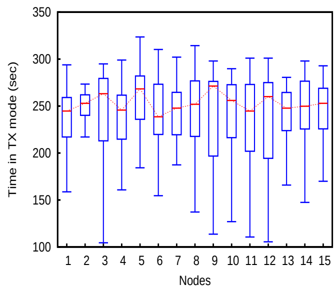
(c) B-MAC: Radio RX Time



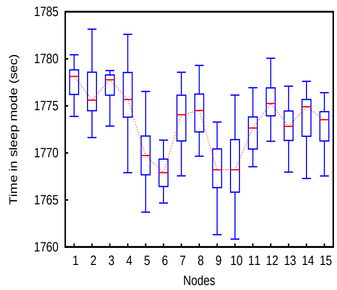
(d) Grid: Radio TX Time



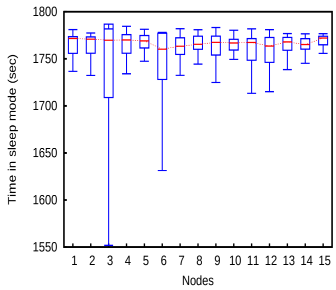
(e) Random: Radio TX Time



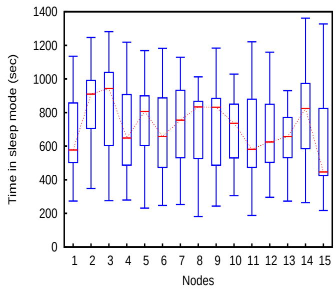
(f) B-MAC: Radio TX Time



(g) Grid: Radio Sleep Time



(h) Random: Radio Sleep Time



(i) B-MAC: Radio Sleep Time

Figure 7.5: A comparison of the radio energy consumption between AREA-MAC and B-MAC protocols for several experiment replications, each lasting for 1800 seconds. Hence the y-axis in each graph shows the total radio receive time (in seconds) out of 1800 seconds for each node. (a), (b), and (c) show the radio receive time, (d), (e), and (f) show the radio transmit time, and (g), (h), and (i) show the radio sleep time for the AREA-MAC grid, AREA-MAC random, and B-MAC random topologies respectively. This figure clearly demonstrates that AREA-MAC achieves significant energy gain over B-MAC protocol.

AREA-MAC nodes) in sending long preambles for each and every data packet they receive, as depicted in Figure 7.5f.

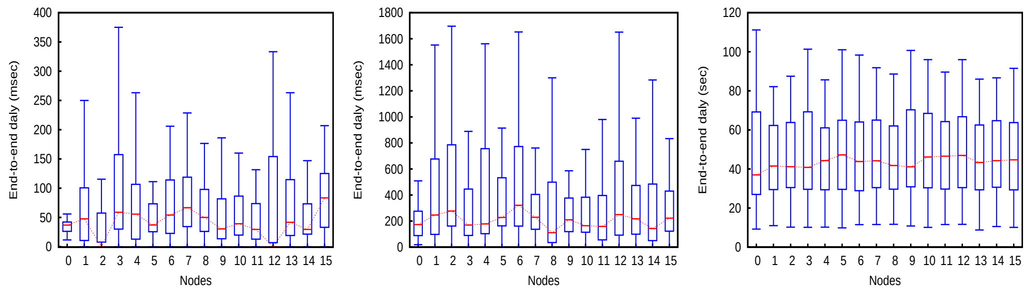
Figures 7.5g, 7.5h, and 7.5i establish a comparison of the radio sleep time between all three considered networks. AREA-MAC protocol achieves ultra low duty cycle for nodes (on average, 1.25% with the grid topology and 1.9% with the random topology) by putting them in sleep mode for a higher time. Border nodes under the AREA-MAC grid topology remain awake for a marginally higher time than nodes that are at the single-hop distance from the sink node. However, due to the planned node location, the grid topology results in better radio sleep times than the random topology for AREA-MAC protocol. B-MAC protocol offers only up to 30% duty cycle for nodes. Hence AREA-MAC offers an enormous amount of energy saving compared to B-MAC protocol for a normal traffic scenario and smaller network size.

7.2.1.2 End-to-End Delay

The end-to-end delay of the received data packets for all three considered network variants is depicted in Figure 7.6. We are mainly concerned for the end-to-end delay at the sink node. The delay for B-MAC nodes in Figure 7.6c is substantially higher (that is why it is measured in sec) than that of the AREA-MAC nodes, which is measured in msec, due to two obvious reasons. First, B-MAC nodes send a long and extended preamble prior to data transmission, irrespective to the wake up time of the intended receiving node. Second, B-MAC nodes broadcast data packets, and the long preamble of at least the size of the check interval duration enables each of the neighboring node to receive the preamble and the subsequent (and probably redundant) data packet. Hence B-MAC nodes receive and process more data packets than AREA-MAC nodes, and in result to that the queuing delay at each node increases, which in return sharply increases the end-to-end packet delay for B-MAC nodes.

On contrary, AREA-MAC offers immensely reduced packet delay than B-MAC protocol, especially for the grid topology (Figure 7.6a), where relatively more nodes are at the single-hop distance from the sink node. For the AREA-MAC grid topology, the delay is about 4 times better than the AREA-MAC random topology and is several times better than B-MAC protocol. AREA-MAC nodes under the random topology usually need multi-hop communication for their packets to arrive at the sink node, and hence they incur marginally higher packet delays compared to the grid topology, as drawn in Figure 7.6b.

On a whole, short preambles, adaptive duty cycling, and short sleep features of AREA-MAC protocol enormously improve the end-to-end packet delay as compared to B-MAC protocol.



(a) Grid: E2E Delay (b) Random: E2E Delay (c) B-MAC: E2E Delay

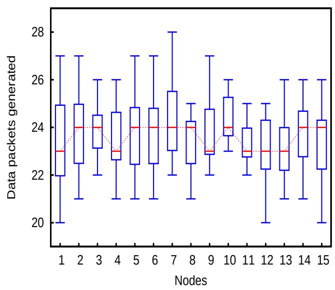
Figure 7.6: A comparison of the end-to-end packet delay between (a) AREA-MAC protocol with the grid topology, (b) AREA-MAC protocol with the random topology, and (c) B-MAC protocol with the random topology. Note that the delay for (a) and (b) is measured in milliseconds, and for (c) it is measured in seconds, which distinctly points the substantial delay gain of AREA-MAC protocol over B-MAC protocol.

7.2.1.3 Packet Delivery Ratio

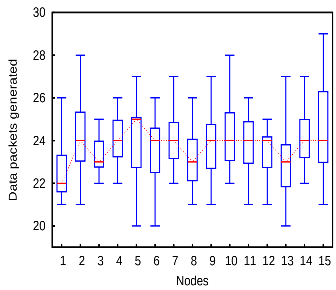
In Figure 7.7, we present an analysis of data packet transmission and reception of nodes for all three network variants. This analysis includes data packet generation, data packet transmission, data packet reception, and the packet delivery ratio at the sink node. Since the sink node does not generate and send data packets, we do not include it in the data generating and transmitting evaluation.

Figures 7.7a, 7.7b, and 7.7c demonstrate the number of data packets generated by the application layer of each node for the AREA-MAC grid topology, AREA-MAC random topology, and B-MAC random topology respectively. Nodes of all three networks generate data packets with the same interval and deviation values, therefore they generate, more or less, a constant number (between 20 to 29) of data packets for an experimentation run of 1800 seconds.

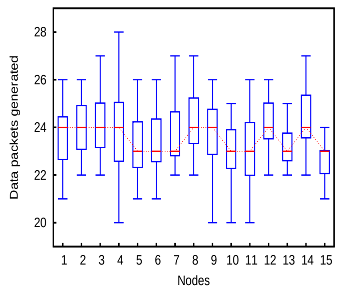
Since we consider a multi-hop scenario through out this evaluation and comparison of AREA-MAC and B-MAC protocols, all nodes receive and forward data packets until packets are arrived at the sink node. The total number of data packets transmitted by a node, which includes both packets generated by the node itself and packets received from other nodes, is indicated in Figures 7.7d, 7.7e, and 7.7f for the AREA-MAC grid, AREA-MAC random, and B-MAC random topologies respectively. As anticipated, B-MAC nodes receive and hence forward a higher number of data packets due to the broadcasting of both preambles and data packets. AREA-MAC sending nodes though broadcast preambles, they forward data packets only towards the nodes that have sent a pre-ACK frame to the sender. Therefore, nodes under both grid and random topologies of AREA-MAC network transmit a less number of data packets (less than 4 times of B-MAC protocol). AREA-MAC nodes with the random topology, however, transmit more packets than the grid topology because more nodes are at the multi-hop distance from the



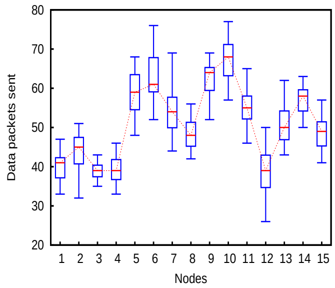
(a) Grid: Generated Packets



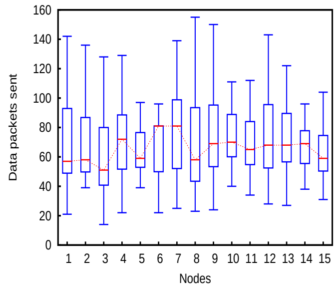
(b) Random: Generated Packets



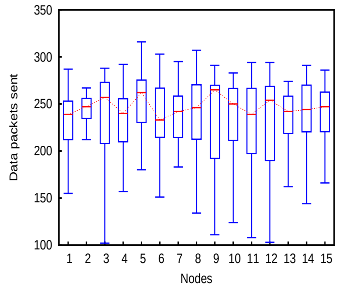
(c) B-MAC: Generated Packets



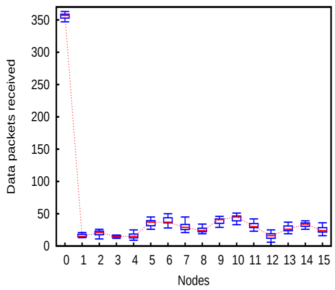
(d) Grid: Sent Packets



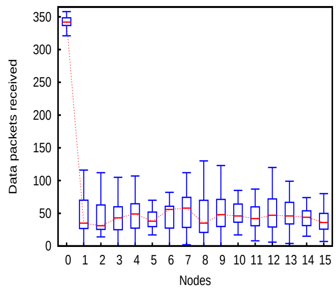
(e) Random: Sent Packets



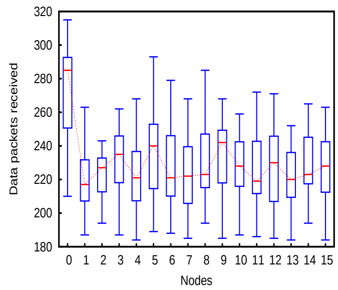
(f) B-MAC: Sent Packets



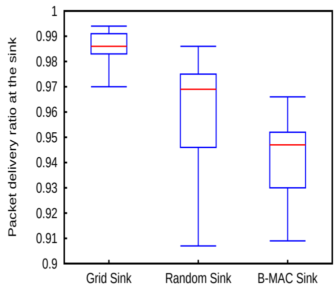
(g) Grid: Recd. Packets



(h) Random: Recd. Packets



(i) B-MAC: Recd. Packets



(j) Packet Delivery Ratio

Figure 7.7: A comparison of the packet delivery ratio including data packet generation, transmission, and reception between AREA-MAC and B-MAC protocols. (a), (b), and (c) present the data packet generated, (d), (e), and (f) present the data packet transmitted, and (g), (h), and (i) present the data packet received by nodes of the AREA-MAC grid, AREA-MAC random, and B-MAC random topologies respectively. (j) compares the overall packet delivery ratio at the sink node for all three networks.

sink node, therefore they receive (and then forward) more packets. Central nodes of the grid topology, i.e., nodes with ID ranges between 5 - 7 and 9 - 11 transmit a higher number of data packets as they behave as a gateway for the farthest nodes from the sink node, i.e., nodes with ID ranges between 13 - 15.

Received data packets by each node including the sink are depicted in Figures 7.7g, 7.7h, and 7.7i for all three networks. Again, we are mostly interested for the higher number of (non-duplicate) data packets that successfully arrive at the sink node. At the same time, other nodes must not receive too many data packets, especially the duplicate or already treated ones. The higher the number of packets a node receives, the larger is its energy consumption. This is exactly the case with B-MAC protocol, where all normal nodes receive a large number of data packets (more than 4 times of AREA-MAC normal nodes); most of them are duplicate ones. This number is very close to the number of data packets that are received by the B-MAC sink node. On contrary, the AREA-MAC features enable its sink node to receive much higher number of data packets than its normal nodes. This number is, however, slightly higher for the AREA-MAC grid topology than the AREA-MAC random topology because the sink node of the grid has more direct neighbors. Alternatively, the normal nodes of the AREA-MAC random topology receive slightly more data packets than the normal nodes of the AREA-MAC grid topology. Understandably, the central nodes of the AREA-MAC grid receive more data packets than the border nodes.

The overall packet delivery ratio at the sink node for all three networks is depicted in Figure 7.7j. As expected, the AREA-MAC grid topology offers better packet delivery ratio, which almost touches 100%, than other considered networks. The AREA-MAC random topology provides a little reduction in this value, which is further reduced under B-MAC protocol. On a whole, all three variants offer an acceptable packet delivery ratio, however B-MAC nodes consume more energy and time to achieve this mark.

7.2.1.4 Preamble Transmission and Reception

In Figure 7.8, we compare the number of preambles that nodes (except the sink node) of all three networks transmit and receive. As said, B-MAC protocol operates under the long preamble scheme, where nodes send out a long preamble prior to each data packet. Since it is not possible for packet-based radios to send out a long preamble, we substitute the long preamble by a series of continuous short preambles that nodes transmit for a complete check interval duration. As nodes in a WSN usually generate data packets at a low rate, preambles being control packets consume more radio energy. Therefore, in an energy efficient WSN, nodes must reduce the number of preambles they transmit or receive. It is obvious from Figures 7.8a, 7.8b, and 7.8c, B-MAC nodes transmit a substantially higher number of preambles than AREA-MAC nodes due to the transmission of long preambles. Nevertheless, due to multi-hop transmission, border nodes of the AREA-MAC grid topology transmit (almost 6 times) more preambles than other nodes, which are single-hop away from the sink node. Nodes of the AREA-MAC random topology transmit relatively more preambles than grid nodes as per their distance from the sink node.

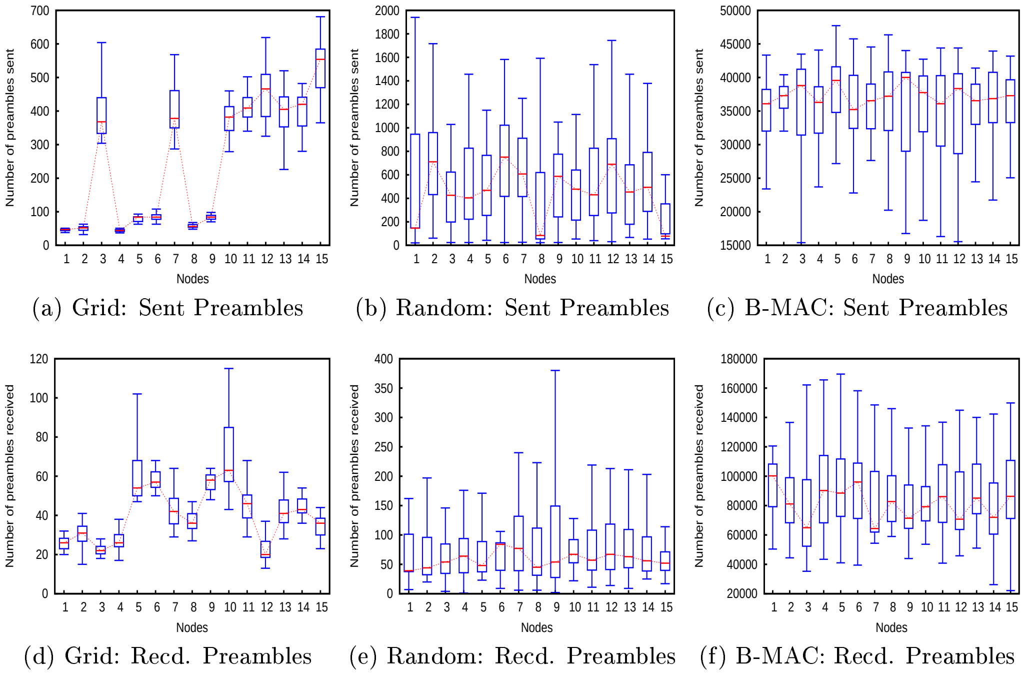
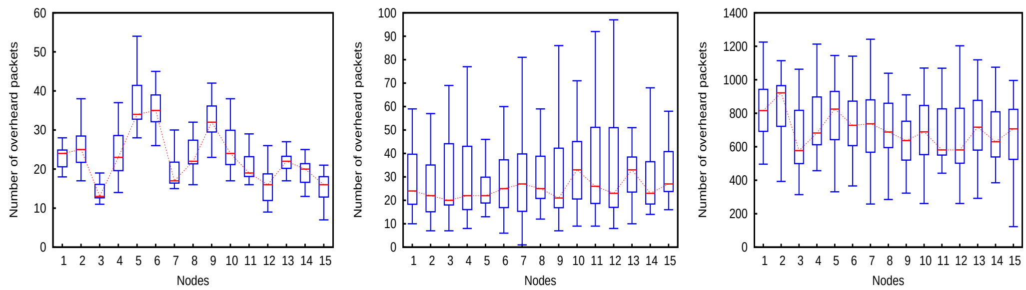


Figure 7.8: A comparison of the preamble transmission and reception between AREA-MAC and B-MAC protocols. (a), (b), and (c) depict the number of preambles sent and (d), (e), and (f) depict the number of preambles received by nodes of the AREA-MAC grid, AREA-MAC random, and B-MAC random topologies respectively.

The number of preambles received by nodes of all three networks is given in Figures 7.8d, 7.8e, and 7.8f. On finding a preamble on the channel during the wake-up interval, B-MAC nodes stay in receive mode and continue to receive the preamble until the subsequent data packet arrives. Consequently, they pick up a significantly large number of preambles than AREA-MAC nodes, which immediately send out a pre-ACK frame once they receive a preamble. For the AREA-MAC grid topology, nodes that are nearer to the sink node receive almost half the number of the preambles that the border nodes receive because the sink node being in constant receive mode acquires most of preambles available in its surroundings. On contrary, all the AREA-MAC nodes working under the random topology receive, more or less, an equal number of preambles. These nodes receive and transmit almost the same number of preambles as the border nodes of the AREA-MAC grid topology do.

7.2.1.5 Overheard Packets

As mentioned in Section 2.4.2, overheard packets constitute a main portion of energy waste in WSNs. Figure 7.9 calculates the number of overheard packets received by nodes for all three considered networks. For nodes of both AREA-MAC grid and random topologies, an overheard packet can be a data packet, preamble, or pre-ACK frame depending on the node state in which the frame is received. For B-MAC protocol, where nodes broadcast both data packets and preambles, we consider only duplicate packets (packets that have already been treated by the node) and loop back packets (packets that have been generated by the node itself) as overheard packets. Figures 7.9a, 7.9b, and 7.9c depict the number of overheard packets for nodes of the AREA-MAC grid, AREA-MAC random, and B-MAC random topologies respectively.



(a) Grid: O-heard Packets (b) Random: O-heard Packets (c) B-MAC: O-heard Packets

Figure 7.9: A comparison of overheard packets between nodes of the (a) AREA-MAC grid topology, (b) AREA-MAC random topology, and (c) B-MAC random topology. Due to the transmission of long preambles and broadcasting of data packets, B-MAC nodes receive a substantially higher number of overheard packets than AREA-MAC nodes.

AREA-MAC nodes under both topologies receive much (up to 30 to 40 times) less number of overheard packets than B-MAC nodes. For the AREA-MAC grid topology, nodes that are nearer to the sink node receive slightly more overheard packets as they receive more frames including preambles, data, or pre-ACK frames due to the funneling effect of a WSN [88]. All nodes of the AREA-MAC random topology receives, on average, an equal number of overheard packets. Hence, AREA-MAC nodes working under either of the topology preserve more energy by reducing the number of overheard packet that they receive compared to B-MAC nodes.

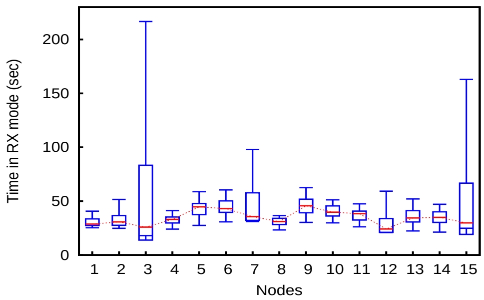
7.2.2 Burst Traffic Scenario

For the burst traffic scenario, we keep most of the simulation configuration of the normal traffic scenario given in Section 7.2.1 same except the data burst size of a node. In order to evaluate the effect of higher traffic, the data burst size has been increased from one to two in this section. It means, for a given experiment time, each of the normal node generates double amount of data packets than the simple traffic scenario. We present various results for energy efficiency, end-to-end delay, data packet transmission and reception, preamble transmission and reception, and overheard packets for both grid and random topologies of an AREA-MAC network for the burst traffic scenario.

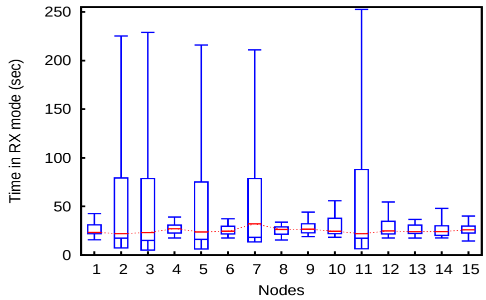
In the previous analysis, we have observed that AREA-MAC protocol outperforms B-MAC protocol for all the considered metrics. It is obvious that the performance of B-MAC protocol would get worse with the increasing number of data packets on the channel because the transmitting nodes need to send a higher number of extended preambles and in result to that all the neighboring nodes receive more preambles and data packets. Hence transmitting, receiving, and overhearing nodes consume a considerable amount of energy. The packet delay of B-MAC nodes also increases with the increasing number of long preambles and (mostly redundant) data packets on the channel. Due to these apparent reasons, we do not add B-MAC protocol for the burst traffic scenario but compare the two different topologies of AREA-MAC protocol. We slightly move the position of the sink node of the AREA-MAC random topology from the left bottom corner (Figure 7.4b) towards the center of the playground so that few more nodes could fall within the single-hop distance from the sink node. This improves the evaluation quality of nodes for the AREA-MAC random topology.

7.2.2.1 Energy Consumption

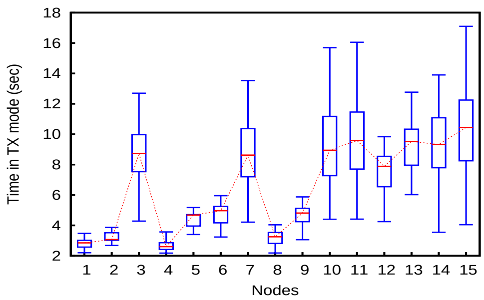
In Figure 7.10, we demonstrate the energy related evaluation of AREA-MAC nodes of grid and random topologies for the burst traffic scenario. Figures 7.10a and 7.10b present the radio receive time for nodes of both networks. Understandably, nodes receive almost double amount of data packets than the simple traffic scenario. However, due to the adaptive duty cycling feature of AREA-MAC, nodes under the burst traffic scenario spend slightly higher time (not the double) in receive mode than the normal traffic scenario (Figures 7.5a and 7.5b). For the grid topology, border nodes that are nearer to the sink node, i.e., nodes with IDs 1, 2, 4, and 8 consume relatively less time in receive mode than other nodes as they receive relatively less number of data packets. Nodes with IDs 3, 7, and 15 have some higher peaks in the radio receive time because they receive more data packets for some experimentation runs. Nevertheless, their receive time remains, on average, same as of other nodes. For the AREA-MAC random topology, all nodes spent more or less same amount of time (about 40 seconds) in receive mode. Few nodes though have some higher radio receive time due to their random locations.



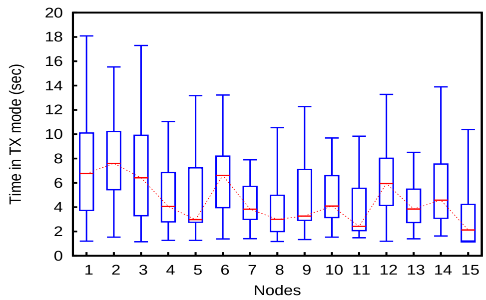
(a) Burst-Grid: Radio RX Time



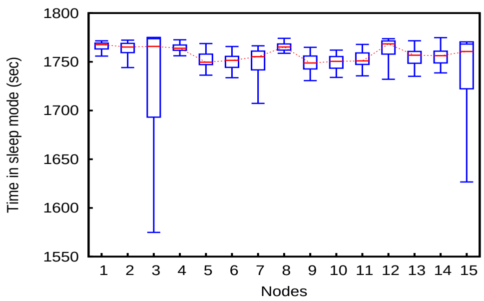
(b) Burst-Random: Radio RX Time



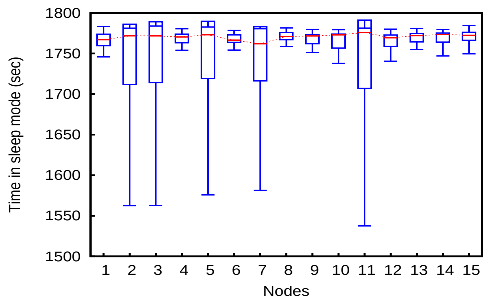
(c) Burst-Grid: Radio TX Time



(d) Burst-Random: Radio TX Time



(e) Burst-Grid: Radio Sleep Time



(f) Burst-Random: Radio Sleep Time

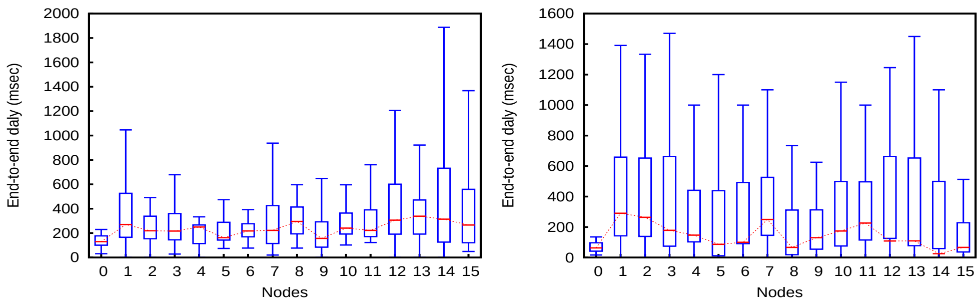
Figure 7.10: A comparison of the radio energy consumption between AREA-MAC grid and random topologies for the burst traffic scenario. (a) and (b) show the radio receive time, (c) and (d) show the radio transmit time, and (e) and (f) show the radio sleep time for nodes of the AREA-MAC grid and random topologies respectively.

The radio transmit time of nodes for the AREA-MAC grid and random topologies for the burst traffic scenario is shown in Figures 7.10c and 7.10d respectively. Like the simple traffic scenario, radios of the border nodes of the AREA-MAC grid topology expend more time in transmit mode than other nodes because they are multi-hop away from the sink node, and therefore they need to send more preambles to get response from their duty-cycled neighbors. Similarly, the radio transmit time of nodes under the AREA-MAC random topology depends on their location and distance from the sink node. On average, nodes of both topologies consume slightly higher time in transmit mode than the normal traffic scenario (Figures 7.5d and 7.5e) due to the transmission of more data packets (and hence more preambles). However, the adaptive duty cycling feature of AREA-MAC helps nodes to keep this difference at minimum.

Figures 7.10e and 7.10f present the radio sleep time for nodes of both AREA-MAC topologies. The grid topology offers slimly better sleep time than the random topology due to the planned deployment of nodes. Nonetheless, AREA-MAC protocol offers quite acceptable duty cycle values (around 2%) for nodes of either topology even under the burst traffic conditions. These values are in fact much better than B-MAC nodes working under the simple traffic scenario (Figure 7.5i).

7.2.2.2 End-to-End Delay

The end-to-end packet delay for nodes of both AREA-MAC topologies for the burst traffic scenario is depicted in Figure 7.11. The delay at the sink node of the grid topology (Figure 7.11a) has been increased by few (up to 150) milliseconds than that of the normal traffic scenario (Figure 7.6a) due to the transmission and reception of more data packets. However, the slight shifting in the location of the sink node has almost halved the packet delay under the random topology for the burst traffic scenario in Figure 7.11b as compared to the simple random



(a) Burst-Grid: E2E Delay (b) Burst-Random: E2E Delay

Figure 7.11: The end-to-end packet delay analysis for AREA-MAC nodes working under the (a) grid topology and (b) random topology for the burst traffic scenario. The delay for AREA-MAC nodes, particularly at the sink node, is much lower than the B-MAC nodes working under low traffic conditions (Figure 7.6c).

topology in Figure 7.6b. These results interpret that the burst traffic does not significantly affect end-to-end packet delay for AREA-MAC nodes, whereas the delay for B-MAC nodes would increase sharply with the increase in traffic. Thus the AREA-MAC features supporting timeliness in a WSN (discussed in Section 6.1.4.2) perform as per expectations.

7.2.2.3 Packet Delivery Ratio

Figure 7.12 depicts data packet related evaluation for nodes of both AREA-MAC grid and random topologies for the burst traffic scenario. Figures 7.12a and 7.12b point out the number of data packets generated by each node for both topologies. Since same data generating and deviation intervals have been used, nodes generate data packets almost the same way as with the simple traffic scenario. However, the burst size value of two results in almost double amount of data packets that a node generates for the burst traffic scenario as compared to the simple traffic scenario (Figure 7.7).

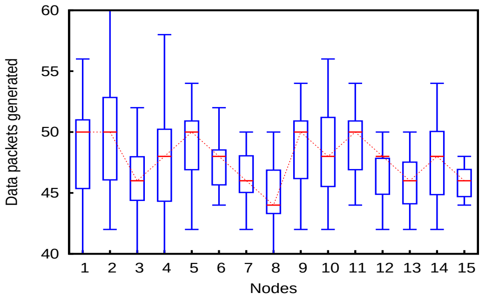
The number of data packets transmitted by nodes of both AREA-MAC grid and random topologies is shown in Figures 7.12c and 7.12d respectively. Due to the burst packet generation, nodes under both topologies send out almost double amount of data packets than the simple traffic scenario, which is depicted in Figures 7.7d and 7.7e. The slight shifting of the sink node though reduces the relative number of transmitted data packets for the AREA-MAC random topology because few more nodes are now within the single-hop vicinity of the sink node.

Figures 7.12e and 7.12f depict the number of data packets received by nodes including the sink node for the burst traffic scenario. Nodes understandably receive more data packets, particularly the sink node that receives almost double amount of data packets, under the burst traffic scenario as compared to the normal traffic scenario. However, the number of data packets received by the normal AREA-MAC nodes is still several time less than B-MAC nodes of the normal traffic scenario (Figure 7.7i), hence they consume much less energy even under high traffic.

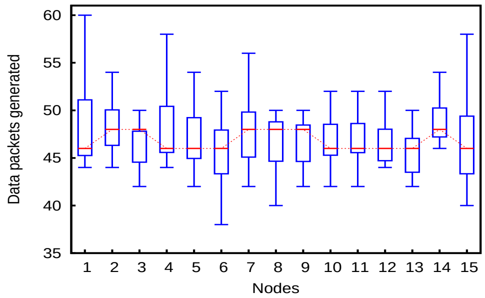
The packet delivery ratio at the sink node for both topologies for the burst traffic scenario is presented in Figure 7.12g. Like the normal traffic scenario, the delivery ratio of the AREA-MAC network is nearly touching 100% even under the burst traffic. Hence, small sized preambles and adaptive duty cycling help AREA-MAC nodes to achieve nearly perfect delivery ratio with the minimum (extra) consumption of energy.

7.2.2.4 Preamble Transmission and Reception

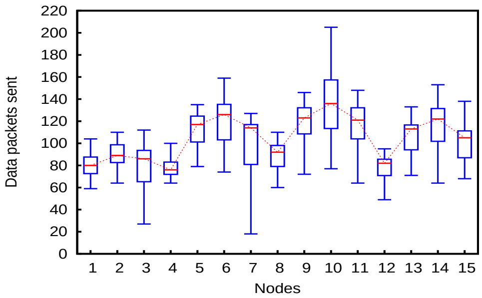
Figure 7.13 compares the number of preambles transmitted and received by nodes of both AREA-MAC grid and random topologies. Though nodes generate double amount of data packets with the burst scenario, the adaptive duty cycling enables them not to transmit and receive double amount of preambles as compared to the simple traffic scenario. Thus, the number of preambles sent (Figures 7.13a



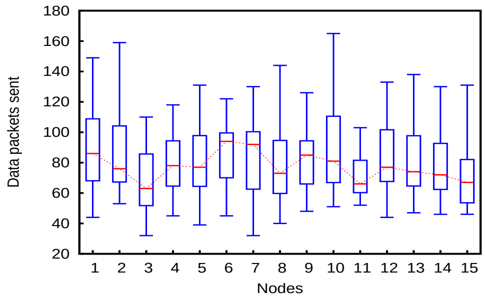
(a) Burst-Grid: Generated Packets



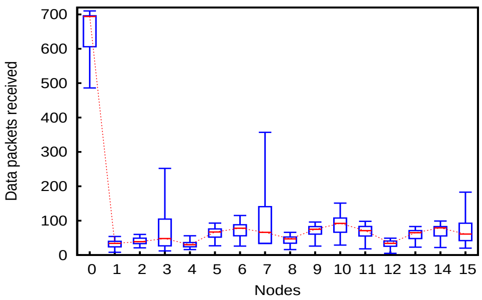
(b) Burst-Random: Generated Packets



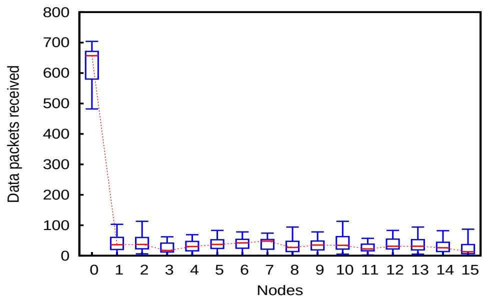
(c) Burst-Grid: Sent Packets



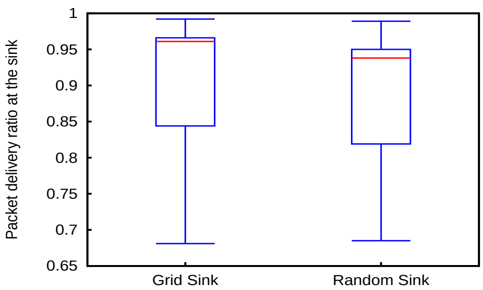
(d) Burst-Random: Sent Packets



(e) Burst-Grid: Recd. Packets

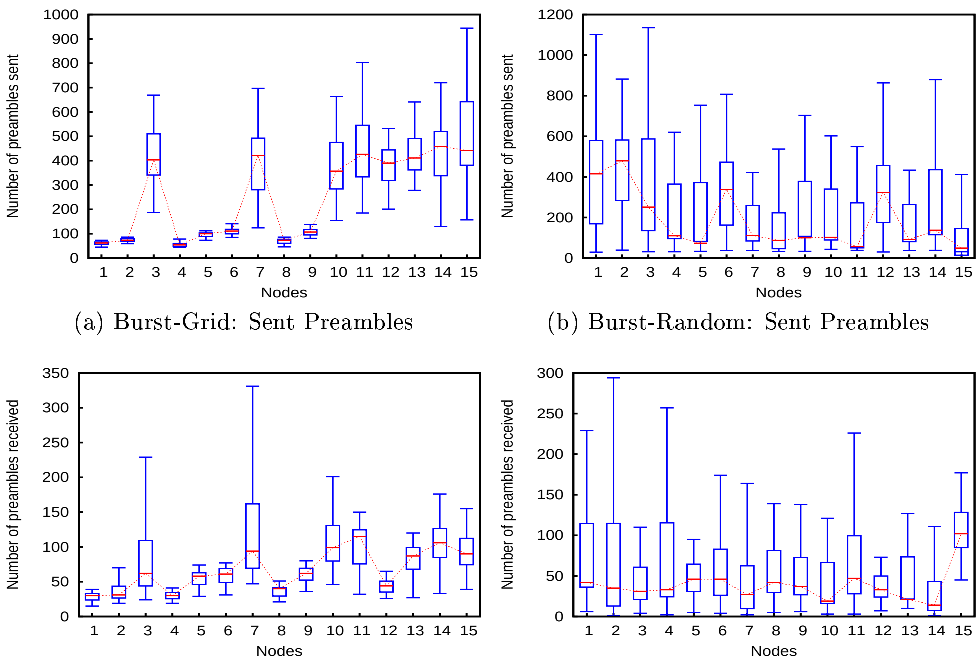


(f) Burst-Random: Recd. Packets



(g) Burst-Packet Delivery Ratio

Figure 7.12: An analysis of the packet delivery ratio including data packet generation, transmission, and reception for nodes of the grid and random topologies of AREA-MAC protocol for the burst traffic scenario. (a) and (b) present the data packet generated, (c) and (d) present the data packet transmitted, and (e) and (f) present the data packet received by nodes of the AREA-MAC grid and random topologies respectively. (g) compares the overall packet delivery ratio at the sink node for both topologies.



(a) Burst-Grid: Sent Preambles

(b) Burst-Random: Sent Preambles

(c) Burst-Grid: Received Preambles

(d) Burst-Random: Received Preambles

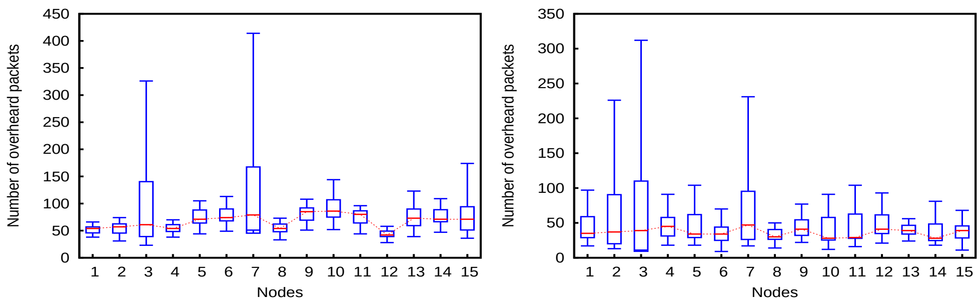
Figure 7.13: A comparison of the preamble transmission and reception for nodes of the AREA-MAC grid and random topologies for the burst traffic scenario. (a) and (b) show the number of preambles sent and (c) and (d) show the number of preambles received by nodes of the AREA-MAC grid and random topologies respectively.

and 7.13b) and received (Figures 7.13c and 7.13d) by AREA-MAC nodes of both grid and random topologies respectively under the burst traffic scenario is much less than the (anticipated double number of) preambles sent and received by nodes with the simple traffic scenario, which is given in Figure 7.8. Border nodes of the AREA-MAC grid topology, which are multi-hop away from the sink node, expectedly transmit and receive a higher number of preambles than the nodes that are at the one-hop distance from the sink node. The number of preambles sent and received by nodes of the random topology mainly depends on their location.

Hence, the number of preambles sent and received by AREA-MAC nodes does not increase exponentially with the increasing number of data packets. B-MAC nodes, however, would send and receive an immense number of preambles, thereby consume a considerable amount of energy with the increasing traffic.

7.2.2.5 Overheard Packets

The number of overheard packets received by nodes of both AREA-MAC grid and random topologies under the burst traffic conditions are depicted in Figures 7.14a and 7.14b respectively. Nodes receive slightly higher number of overheard packets as compared to the AREA-MAC nodes of the normal traffic scenario (Figure 7.9b) due to the transmission of more data packets and preambles on the channel. However, this number is still several time less than for the B-MAC nodes of simple traffic scenario (Figure 7.9c). Border nodes of the AREA-MAC grid that are nearer to the sink node, i.e., nodes with IDs 1 - 4, 8, and 12 understandably receive less number of overheard packets than other nodes. The random topology of AREA-MAC protocol results in more or less an equal number of overheard packets for nodes of the burst traffic scenario as of the simple traffic scenario due to the reason that the sink node now has few more direct neighbors.



(a) Burst-Grid: Overheard Packets (b) Burst-Random: Overheard Packets

Figure 7.14: The number of overheard packets received by nodes of the (a) AREA-MAC grid topology and (b) AREA-MAC random topology for the burst traffic scenario. The figure proves that the number of overheard packets received by AREA-MAC nodes is much smaller than B-MAC nodes.

7.2.3 High Traffic (HT) Scenario

In order to further analyse the performance of AREA-MAC protocol under varying traffic conditions, we increase the data traffic by changing the data generating and deviation intervals of nodes from 75 and 45 to 25 and 15 respectively in this high traffic scenario. It means that all the nodes (except the sink node) generate almost three times more data packets than with the normal traffic scenario given in Section 7.2.1. Note that with the burst traffic scenario (Section 7.2.2), nodes generate two packets at a (random) time based on the given data generating and deviation intervals, whereas with the high traffic scenario, nodes generate a single data packet at a random time but with a higher frequency, i.e., with the lower data generating and deviation intervals.

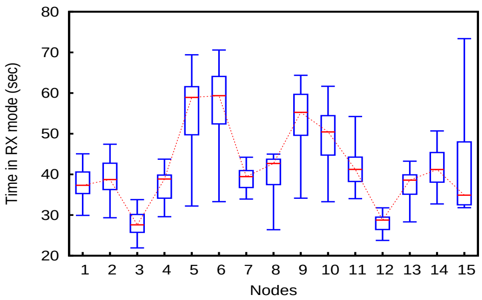
We evaluate the performance of AREA-MAC protocol for energy efficiency, end-to-end delay, data packet transmission and reception, packet delivery ratio, preamble transmission and reception, and overheard packets metrics for both grid and random topologies under this scenario. Like the burst traffic scenario, we slightly move the position of the sink node from the left bottom corner (Figure 7.4b) towards the center of the playground, so that few more nodes could fall within the single-hop distance from the sink node.

7.2.3.1 Energy Consumption

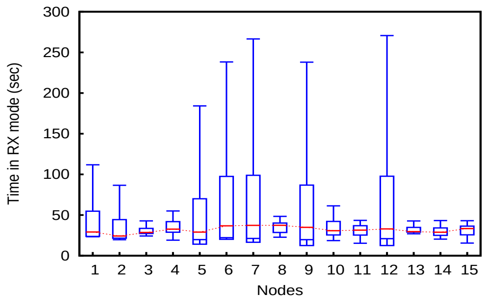
Figure 7.15 presents the energy analysis for nodes of both AREA-MAC grid and random topologies for the high traffic scenario. Figures 7.15a and 7.15b demonstrate the radio receive time for nodes of grid and random topologies respectively. Since nodes generate and process almost three times more data packets than the normal traffic scenario, their radios understandably consume a higher time in receive mode. However, nodes of both topologies behave more or less in the same way as with the respective topology of the normal traffic scenario (Figures 7.5a and 7.5b). Central nodes of the grid topology, particularly nodes with IDs 5 - 6, and 9 - 10 consume more time in receive mode than other nodes because they have more direct neighbors, and therefore they receive and process more preambles and data packets. Alternatively, border nodes of the grid topology have less number of direct neighbors, and therefore they receive relatively a less number of preambles and/or data packets. On the other hand, nodes with the AREA-MAC random topology consume, on average, an equal time in receive mode.

The radio transmit time of nodes for the AREA-MAC grid and random topologies for the high traffic scenario is shown in Figures 7.15c and 7.15d respectively. Since traffic has been increased in this scenario, nodes transmit more data packets and preambles, and therefore they consume more time in transmit mode than the simple and burst traffic scenarios. However, moving the sink node of the random topology towards the center of the playground has reduced this effect. Like both normal and burst traffic scenarios, border nodes of the grid topology for this high traffic scenario spend more time in transmit mode than other nodes because they are multi-hop away from the sink node and need to send more preambles to transmit/forward data packets. Again, the adaptive duty cycling feature of AREA-MAC helps nodes to keep this gap at minimum.

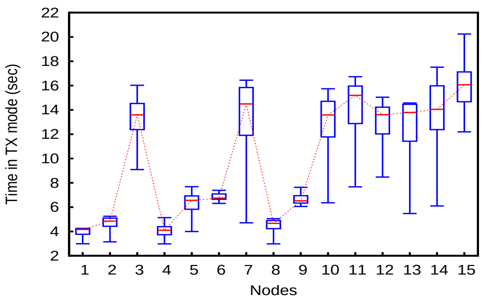
The radio sleep time for nodes of both AREA-MAC grid and random topologies is outlined in Figures 7.15e and 7.15f respectively. Like the burst traffic scenario, the grid topology offers slightly better sleep time than the random topology due to the planned deployment of nodes. Though nodes generate and process a higher number of data packets (and preambles) with the high traffic scenario, AREA-MAC still manages to provide very reasonable duty cycle (between 2% to 3%) for nodes. These duty cycle values are, in fact, much better than B-MAC nodes working under the normal traffic scenario (Figure 7.5i).



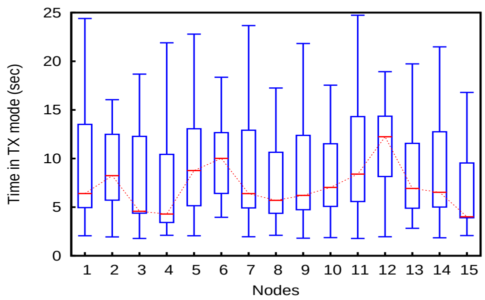
(a) HT-Grid: Radio RX Time



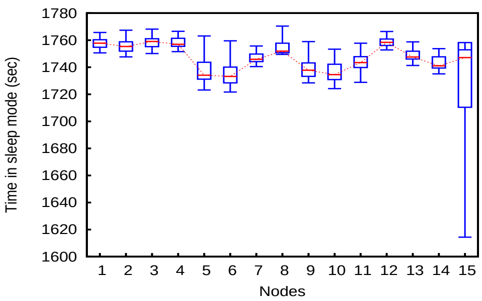
(b) HT-Random: Radio RX Time



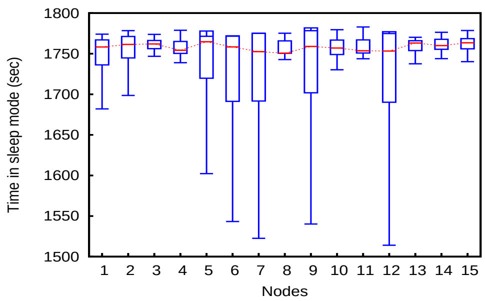
(c) HT-Grid: Radio TX Time



(d) HT-Random: Radio TX Time



(e) HT-Grid: Radio Sleep Time

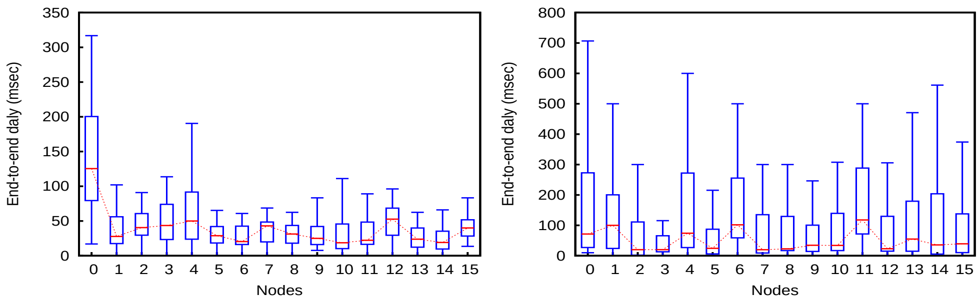


(f) HT-Random: Radio Sleep Time

Figure 7.15: A comparison of the radio energy consumption between AREA-MAC grid and random topologies for the high traffic scenario. (a) and (b) show the radio receive time, (c) and (d) show the radio transmit time, and (e) and (f) show the radio sleep time for nodes of the AREA-MAC grid and random topologies respectively.

7.2.3.2 End-to-End Delay

Figure 7.16 depicts the end-to-end packet delay for nodes of both AREA-MAC topologies for the high traffic scenario. As anticipated, the delay of the grid topology for the high traffic scenario (Figure 7.16a) is higher than that of the normal traffic scenario (Figure 7.6a) and is also marginally higher than the burst traffic scenario (Figure 7.11a) due to the transmission and reception of more data packets. At the same time, the packet delay for the random topology with the high traffic scenario is relatively less than both normal (Figure 7.6b) and burst (Figure 7.11b) scenarios mainly due to the shifting of the sink node towards the center. This evaluation construes that AREA-MAC protocol offers very moderate packet delays even with varying traffic conditions as compared to B-MAC protocol, where the packet delay would grow more or less in an exponential form with the increasing data traffic on the channel.

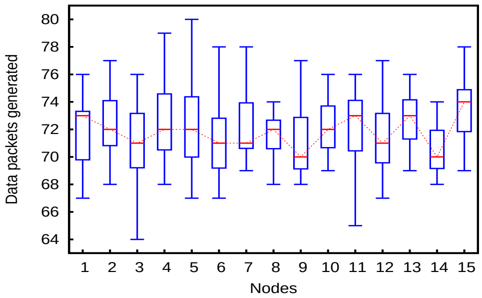


(a) HT-Grid: E2E Delay (b) HT-Random: E2E Delay

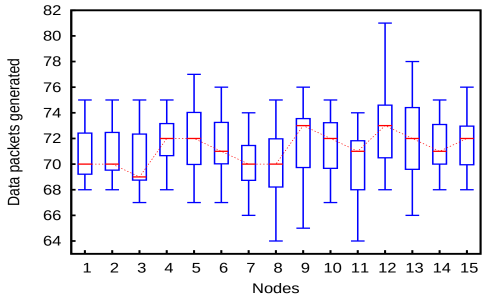
Figure 7.16: A comparison of the end-to-end delay between (a) AREA-MAC protocol with the grid topology and (b) AREA-MAC protocol with the random topology for the high traffic scenario.

7.2.3.3 Packet Delivery Ratio

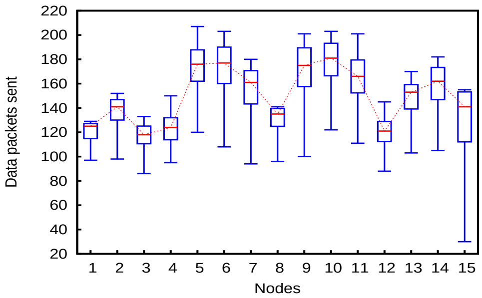
The numbers of data packets generated, received, and transmitted by nodes and packet delivery ratio at the sink node for both AREA-MAC grid and random topologies are depicted in Figure 7.17. As shown in Figures 7.17a and 7.17b, nodes generate about three times more data packets than the normal traffic scenario (Figure 7.7) and 1.5 times more than the burst traffic scenario (Figure 7.12) due to change in the data generating and deviation intervals. Accordingly, nodes under both topologies of the high traffic scenario transmit a higher number of data packets in Figures 7.17c and 7.17d. As expected, central nodes of the AREA-MAC grid topology transmit more data packets than the border nodes because they have more direct neighbors, and hence they receive and forward more packets. Again, the slight shifting of the sink node reduces the relative number of transmitted data packets for nodes of the AREA-MAC random topology, as few more nodes fall into



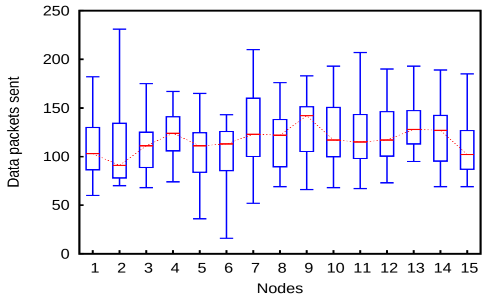
(a) HT-Grid: Generated Packets



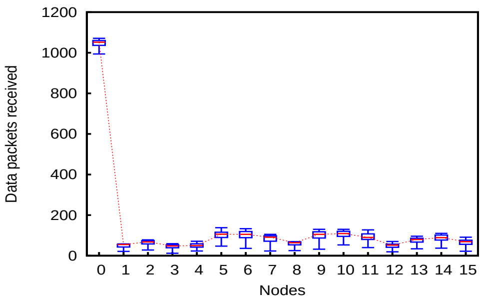
(b) HT-Random: Generated Packets



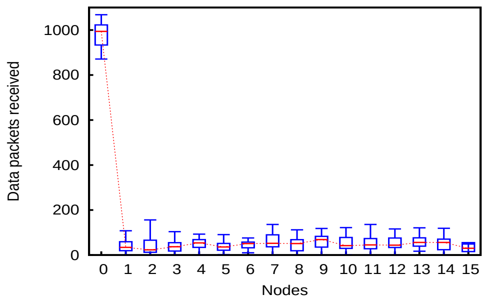
(c) HT-Grid: Sent Packets



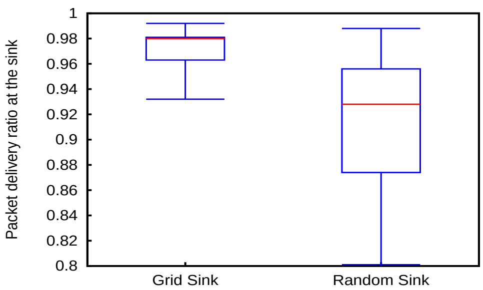
(d) HT-Random: Sent Packets



(e) HT-Grid: Recd. Packets



(f) HT-Random: Recd. Packets



(g) HT-Packet Delivery Ratio

Figure 7.17: An analysis of the packet delivery ratio including data packet generation, transmission, and reception for nodes of the grid and random topologies of AREA-MAC protocol for the high traffic scenario. (a) and (b) present the data packet generated, (c) and (d) present the data packet transmitted, and (e) and (f) present the data packet received by nodes of the AREA-MAC grid and random topologies respectively. (g) compares the overall packet delivery ratio at the sink node for both topologies.

the single-hop vicinity of the sink node.

The numbers of data packets received by AREA-MAC nodes (including the sink node) for both topologies of the high traffic scenario are sketched in Figures 7.17e and 7.17f. Since AREA-MAC protocol targets at the successful delivery of data packets at the sink node, which expectedly and essentially receive a higher number (almost three time the normal scenario and 1.5 times the burst scenario) of data packets under the high traffic conditions. However, normal AREA-MAC nodes still receive (and forward) much less number of data packets as compared to B-MAC protocol (Figures 7.7i), where nodes receive a large number of redundant packets.

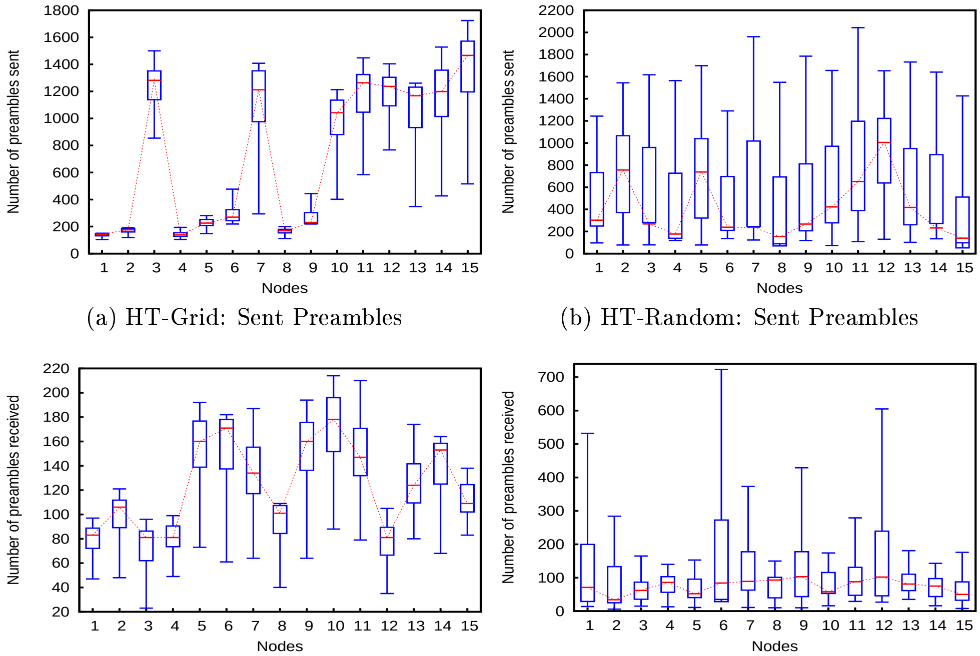
The overall packet delivery ratio at the sink node for both topologies of the high traffic scenario is shown in Figure 7.17g. The grid topology offers, on average, 98% packet delivery ratio, whereas the random topology brings down this figure to around 93%. Though this number is little less than the normal traffic scenario (Figure 7.7j), with the increasing traffic on the channel, it still is a very reasonable figure for a WSN, specially when it is achieved with the minimum energy consumption.

7.2.3.4 Preamble Transmission and Reception

Since nodes under the high traffic scenario generate and process more data packets than other scenarios, they supposedly and inevitably receive and transmit a higher number of preambles as depicted in Figure 7.18. Nodes of the AREA-MAC grid topology transmit and receive almost three times more (and more or less in the same order) preambles in Figures 7.18a and 7.18c respectively with the high traffic scenario than with the grid topology of the normal traffic scenario (Figure 7.8). Nodes that are at the single-hop distance from the sink node transmit significantly less number of preambles than the border nodes, which need to continuously transmit short preambles until their duty-cycled neighbor wake up and acknowledge. Alternatively, the central nodes of the grid topology receive more preambles than the border nodes due to the fact that they receive and forward data packets from the border nodes to the sink node.

Figures 7.18b and 7.18d present the number of preambles sent and received by nodes of the AREA-MAC grid and random topologies respectively for the high traffic scenario. The slight yet worthy displacement of the sink node towards the center of the playground forces the sink node to be in the transmission range of few more nodes, which ultimately decreases the relative number of preambles sent and received by nodes of the random topology. Though this number is higher than both normal traffic (Figure 7.8) and burst traffic (Figure 7.13) scenarios, it is still much less than B-MAC nodes (Figures 7.8c and 7.8f) and is also less for the relative higher number of data packets that nodes generate for the high traffic scenario.

Hence by virtue of the adaptive duty cycling and short preamble features of AREA-MAC protocol, high traffic on the channel does not exponentially increase the relative number of preambles to be sent or received by nodes, as anticipated with B-MAC protocol.



(a) HT-Grid: Sent Preambles (b) HT-Random: Sent Preambles

(c) HT-Grid: Received Preambles (d) HT-Random: Received Preambles

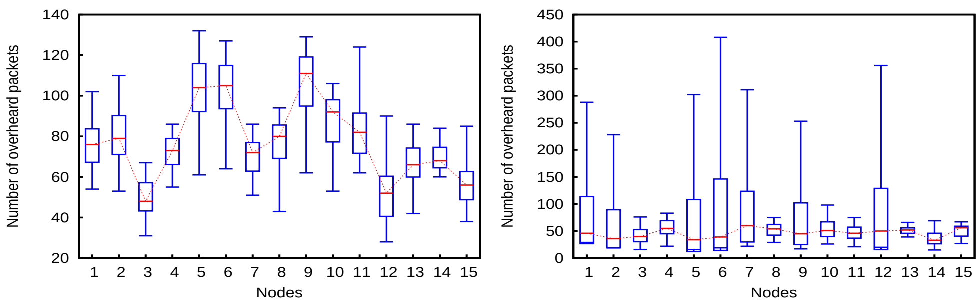
Figure 7.18: A comparison of the preamble transmission and reception for nodes of the AREA-MAC grid and random topologies for the high traffic scenario. (a) and (b) show the number of preambles sent and (c) and (d) show the number of preambles received by nodes of the AREA-MAC grid and random topologies respectively.

7.2.3.5 Overheard Packets

Figures 7.19a and 7.19b respectively exhibit the number of overheard packets that nodes of both AREA-MAC grid and random topologies receive. Due to the presence of higher number of packets (data as well as preambles) on the channel, nodes, particularly of the grid topology, receive more overheard packets as compared to the normal traffic (Figure 7.9) and burst traffic (Figure 7.14) scenarios. Central nodes of the grid topology, i.e., nodes with IDs 5 - 6, and 9 - 10 receive more overheard packets than other nodes for the obvious reason that they have more single-hop neighbors, and therefore they receive more packets.

On the other hand, nodes of the AREA-MAC random topology do not receive a relative higher number of overheard packets even though a higher number of data packets are being transmitted on the channel because of two reasons: 1) the sink node has been shifted towards the center of the playground and 2) the adaptive duty cycling feature of AREA-MAC protocol enables nodes to transmit and receive relatively a lower number of preambles on the channel.

Consequently, AREA-MAC protocol performs acceptably and reasonably well with the varying traffic conditions, whereas the performance of B-MAC protocol



(a) HT-Grid: Overheard Packets (b) HT-Random: Overheard Packets

Figure 7.19: An analysis of the overheard packets that are received by nodes of the (a) AREA-MAC grid topology and (b) AREA-MAC random topology for the high traffic scenario.

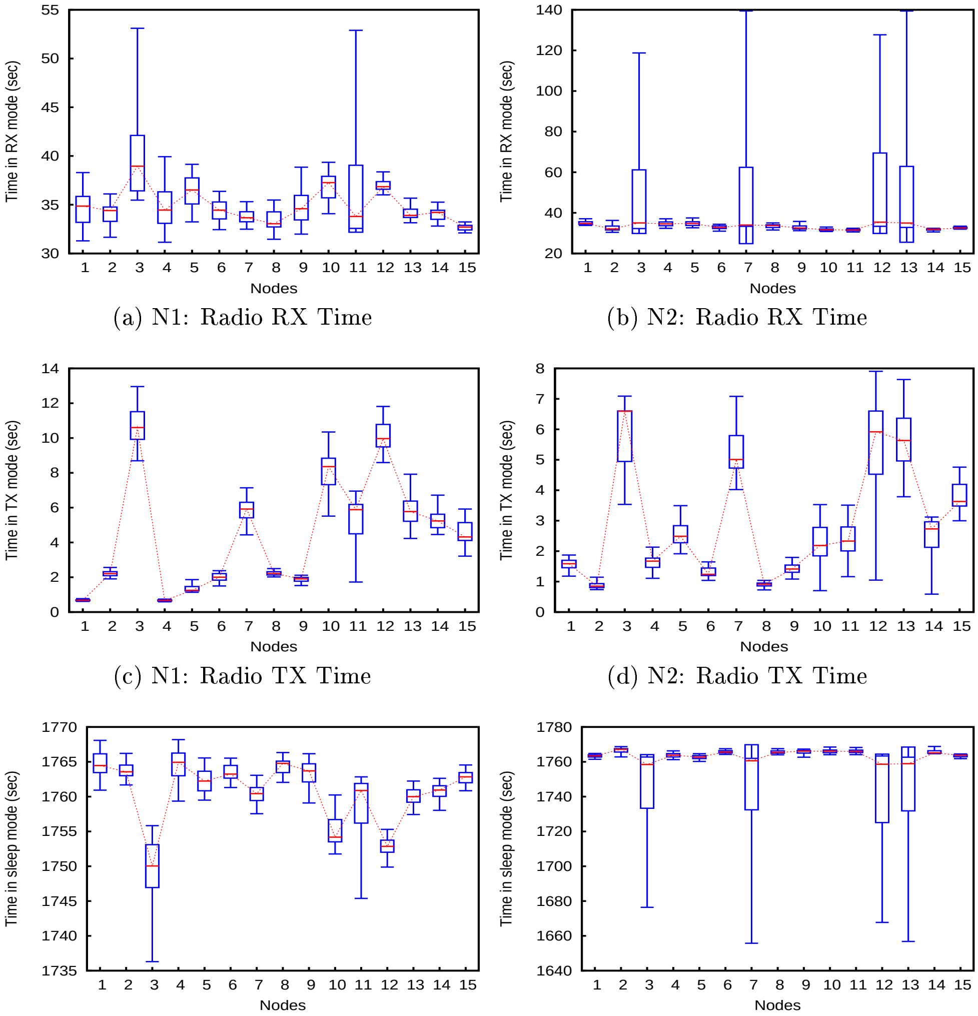
would get worse with the increasing traffic on the channel. Short preambles, adaptive duty cycling, and other AREA-MAC features enable nodes to preserve more radio energy with less packet delay, higher packet delivery, and less control and overheard packets.

7.2.4 Varying Routing Effects

As explained in Sections 4.2.9 and 6.2.3, AREA-MAC considers three types of simple yet effective routing/dissemination schemes, namely N_0, N_1, and N_2 schemes. With the N_0 scheme, nodes broadcast preambles, and any neighbor node irrespective to its location can receive and process the preamble (and the subsequent data packet). With the N_1 scheme, nodes send packets only to their up-level N_1 neighbors. Whereas, nodes with the N_2 routing send packets only to their up-level N_2 neighbors. The evaluation presented in the previous sections implies the N_0 routing scheme. In this section, we discuss the effects of N_1 and N_2 routing schemes on the performance of nodes of the AREA-MAC grid topology by using the simulation configuration given in Section 7.2.1. The main reason behind implementing different types of routing schemes is to check whether the packet delay of AREA-MAC nodes can be improved and/or energy consumption of nodes gets effected with the varying routing schemes.

7.2.4.1 Energy Consumption

In Figure 7.20, we demonstrate energy consumption of AREA-MAC nodes for both N_1 and N_2 schemes. Figures 7.20a and 7.20b depict the radio receive time for both schemes, where nodes deplete relatively higher energy than nodes of the N_0 scheme (Figure 7.5a). The same is also true for the radio transmit time for nodes of both schemes as shown in Figures 7.20c and 7.20d. Hence, nodes with the N_2 scheme consume marginally higher energy than the N_1 scheme, and nodes with



(a) N1: Radio RX Time (b) N2: Radio RX Time

(c) N1: Radio TX Time (d) N2: Radio TX Time

(e) N1: Radio Sleep Time (f) N2: Radio Sleep Time

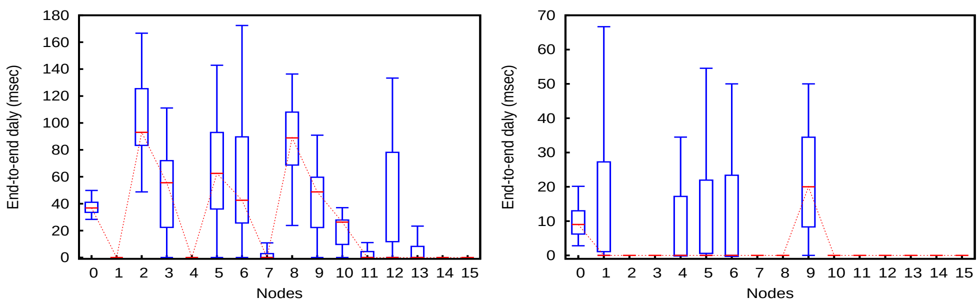
Figure 7.20: An energy analysis of AREA-MAC nodes for the varying routing schemes. (a) and (b) depict the radio receive time, (c) and (d) depict the radio transmit time, and (e) and (f) depict the radio sleep time for AREA-MAC nodes of the N_1 *and* N_2 *schemes respectively. Previously discussed Figures 7.5a, 7.5d, and 7.5g respectively present the receive, transmit, and sleep time of AREA-MAC grid nodes for the* N_0 *scheme.*

the N_1 scheme consume higher energy than the N_0 scheme. The reason behind this distinction is that the sending nodes of the N_1 and N_2 schemes need to send more preambles than the N_0 scheme as neighbor nodes that are neither N_1 nor N_2 up-level neighbors discard received preambles. Since sending nodes transmit a higher number of preambles, receiving nodes receive more preambles, and hence they spent more time in receive mode than the N_0 scheme. Border nodes of both schemes deplete considerably higher energy in transmit mode than other nodes because they are multi-hop away from the sink node, and thus they need to send out many preambles.

The higher time in transmit and receive modes of nodes means their sleep time also reduces, as depicted in Figures 7.20e and 7.20f. Consequently, duty cycle values of nodes with N_1 and N_2 schemes marginally increase as compared to nodes working under the N_0 scheme.

7.2.4.2 End-to-End Delay

Figure 7.21 confirms the improvement in the packet delay for AREA-MAC nodes working under both N_1 and N_2 schemes as compared to the N_0 scheme (Figure 7.6). Since we are mainly concerned for the end-to-end packet delay at the sink node, the N_1 scheme improves the delay slimly better for the sink node in Figure 7.21a. The major delay improvement at the sink node, however, can be observed with the N_2 scheme in Figure 7.21b.



(a) N1: E2E Delay (b) N2: E2E Delay

Figure 7.21: A comparison of the end-to-end packet delay between the (a) AREA-MAC N_1 routing scheme and (b) AREA-MAC N_2 routing scheme. The figure confirms betterment in the packet delay for both N_1 and N_2 schemes, particularly with the N_2 scheme, as compared to the N_0 scheme that has been discussed earlier in this chapter.

For the N_1 scheme, nodes near to the sink node, i.e., nodes with IDs 1 and 4 have minimum delay as the sink node being in constant receive mode gets all data packets in its neighborhood. Node with ID 15 is not a N_1 neighbor for any of the node, whereas node with ID 14 receives very less amount of data packets with the negligible delay.

Border nodes with the N_2 scheme, i.e., nodes with IDs 3, 7, 11 - 15 have negligible delay as they are not N_2 neighbors for any node, and hence they do not receive any data packets. Same is also true for the node ID 10. Though nodes with IDs 2 and 8 are the N_2 neighbors for some nodes, they receive data packet occasionally. Once they receive a packet, they immediately forward it to the sink node, which is their immediate N_2 neighbor. Therefore, both nodes have negligible packet delay.

7.2.4.3 Packet Delivery Ratio

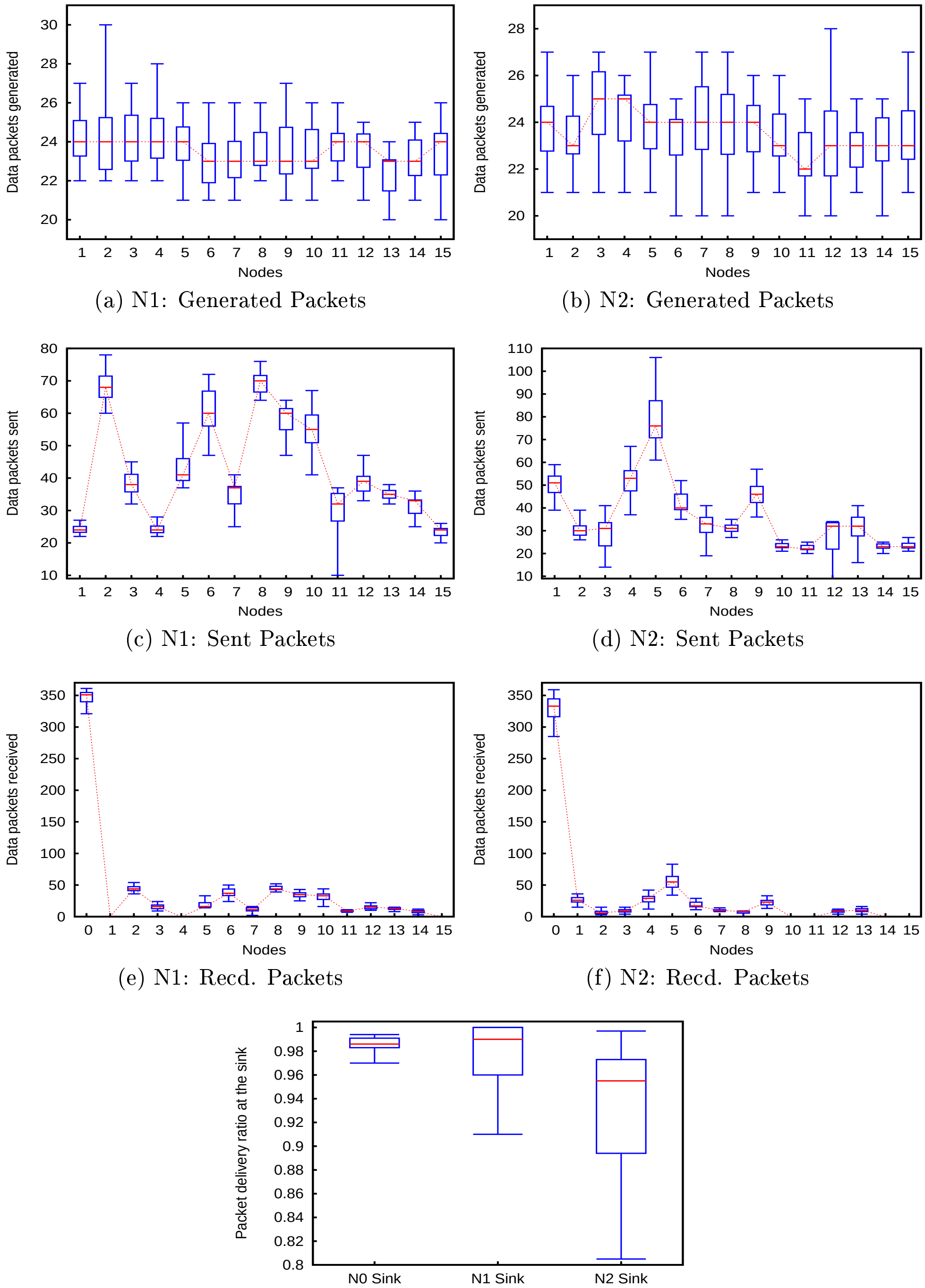
Figure 7.22 outlines the data packet related analysis for both N_1 and N_2 schemes. Since nodes generate data packets with the same data generating and deviation intervals as with the N_0 scheme of the normal traffic scenario (Figure 7.7a), they generate more or less an equal number of data packets in Figures 7.22a and 7.22b for the N_1 and N_2 schemes respectively.

Figures 7.22c and 7.22d depict the number of data packets sent by AREA-MAC nodes under the N_1 and N_2 schemes respectively. With the N_1 scheme, nodes that are nearer to the sink node, i.e., nodes with IDs 1 and 4 transmit less number of data packets, most of them are generated by themselves. They rarely receive packets from their neighbors as the sink node captures most of the surrounding data packets. Similarly node with ID 15 does not receive any packet from its neighbors; it sends out packets that are generated by itself. Central nodes, i.e., nodes with IDs 2, 6, and 8 - 10 understandably transmit a higher number of data packets with the N_1 scheme because they receive packets from their low-level neighbor, which are multi-hop away from the sink node.

With the N_2 routing scheme, border nodes that are not the N_2 neighbors for any node, i.e., nodes IDs from 10 - 11, and 14 - 15 send a minimum number of data packets. Nodes that have minimum number of N_2 low-level neighbors (mostly only one), i.e., nodes with IDs 2 - 3, 7 - 8, and 12 - 13 transmit relatively a higher number of data packets. Other (central) nodes transmit the highest number of data packets.

The number of received data packets for both schemes is calculated in Figures 7.22e and 7.22f. Under the N_1 scheme, nodes that are either nearer to the sink node (i.e., nodes with IDs 1 and 4) or nodes having no low-level neighbors (i.e., node with ID 15) do not receive any data packet. Border nodes with IDs 3, 7, and 11 - 14 receive a small number of data packets, whereas central nodes receive a higher number of data packets. For the N_2 routing scheme, nodes that do not have any N_2 low-level neighbors, do not receive any data packet. Central nodes, i.e., nodes with IDs 1, 4 - 6, and 9 receive relatively a higher number of data packets as they have more N_2 low-level neighbors.

The overall packet delivery ratio at the sink node for all three N_0, N_1, and N_2 routing schemes is compared in Figure 7.22g. Both N_0 and N_1 schemes, on average, receive 99% of the packets that are generated by nodes. The sink node under the N_2 scheme, however, receives relatively a less number of the generated



(a) N1: Generated Packets

(b) N2: Generated Packets

(c) N1: Sent Packets

(d) N2: Sent Packets

(e) N1: Recd. Packets

(f) N2: Recd. Packets

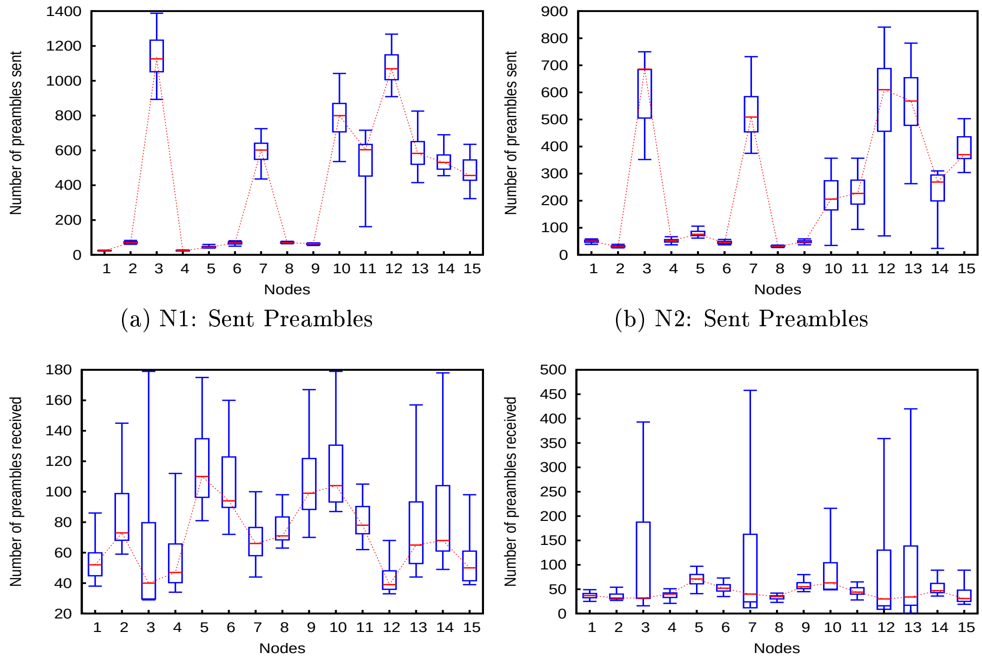
(g) Packet Delivery Ratio

Figure 7.22: An analysis of the packet delivery ratio including data packet generation, transmission, and reception for nodes of the AREA-MAC N_1 *and* N_2 *routing schemes. (a) and (b) present the data packet generated, (c) and (d) present the data packet transmitted, and (e) and (f) present the data packet received by nodes of the AREA-MAC* N_1 *and* N_2 *routing schemes respectively. (g) compares the overall packet delivery ratio at the sink node for all three, i.e.,* N_0, N_1, *and* N_2 *routing schemes.*

packets as the nodes that are not the N_2 up-level neighbors of the sending node do not accept its preambles (and data packets). For a WSN, where nodes are usually densely deployed, neighbor nodes may generate redundant packets (by sensing the same event). Therefore, the achieved delivery ratio under all three N_0, N_1, and N_2 routing schemes of AREA-MAC protocol is quite acceptable. More importantly, the sink node of AREA-MAC protocol largely receives non-redundant data packets, unlike B-MAC protocol, where nodes continuously forward the redundant packets.

7.2.4.4 Preamble Transmission and Reception

In Figure 7.23, we compare the number of preambles that are transmitted and received by AREA-MAC nodes (except the sink node) for the N_1 and N_2 routing schemes. It is obvious from Figures 7.23a and 7.23b that the border nodes of both schemes transmit a substantially higher number of preambles than other nodes that are single-hop away from the sink node. Since nodes with IDs 3 and 12 have only one up-level N_1 neighbors, they send a significantly higher number of preambles with the N_1 scheme than with the N_0 and N_2 schemes.



(a) N1: Sent Preambles

(b) N2: Sent Preambles

(c) N1: Received Preambles

(d) N2: Received Preambles

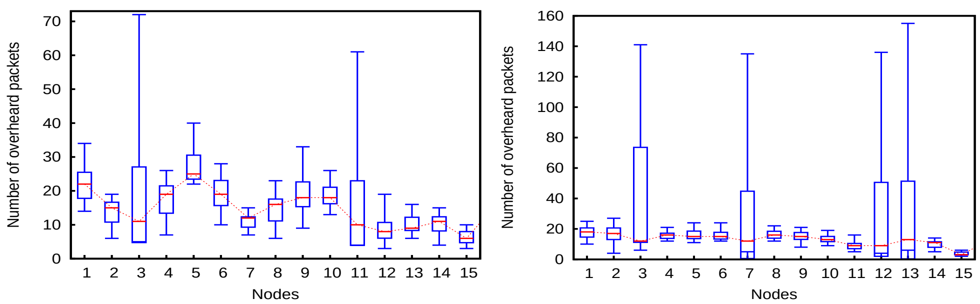
Figure 7.23: An analysis of the preamble transmission and reception for nodes of the AREA-MAC N_1 and N_2 routing schemes. (a) and (b) present the number of preambles sent and (c) and (d) present the number of preambles received by nodes of the N_1 and N_2 schemes respectively.

The number of preambles received by nodes of both N_1 and N_2 schemes is given in Figures 7.23c and 7.23d. With the N_1 scheme, central nodes of the grid receive a higher number of preambles than other nodes. These nodes also receive more preambles than nodes of the N_0 scheme. Nodes nearer to the sink node, i.e., nodes with IDs 1 and 4, nodes having no up-level N_1 neighbors, i.e., node with ID 15, and nodes having minimum up-level N_1 neighbors, i.e., nodes with IDs 3 and 12 receive a less amount of preambles. For the N_2 scheme, central nodes, i.e., nodes with IDs 5-6, and 9-11 receive relatively a higher number of preambles than other nodes. However, average number of received preambles with the N_2 scheme for all nodes remains more or less at the same level.

On a whole, since nodes only accept preambles from their low-level neighbors, transmitting nodes need to send out more preambles with the N_1 and N_2 schemes as compared to the N_0 scheme. In result to that, they also receive more preambles.

7.2.4.5 Overheard Packets

The number of overheard packets received by AREA-MAC nodes for the N_1 and N_2 schemes is outlined in Figure 7.24. Nodes under both schemes receive, on average, same amount of overheard packets as they receive with the N_0 scheme (Figure 7.9). For the N_1 scheme, nodes that are nearer to the sink node, i.e., nodes with IDs 1, 4 - 6, and 8 - 10 receive marginally higher number of overheard packets than other nodes. Nodes with the N_2 scheme receive more or less same number of overheard packets. However, border nodes, i.e., nodes with IDs 3, 7, and 11 - 15 receive slightly less number of overheard packets.



(a) N1: Overheard Packets (b) N2: Overheard Packets

Figure 7.24: The number of the overheard packets that are received by AREA-MAC nodes under the (a) N_1 routing scheme and (b) N_2 routing scheme.

Conclusively, the N_1 and N_2 routing schemes perfectly establish the trade-off between energy and packet delay. Though both schemes marginally increase the energy consumption of AREA-MAC nodes, they also offer improved packet delay at the sink node. Hence such schemes can be used with the strict delay bound WSN applications.

7.2.5 Scalability Effect - A Medium Size Network

In order to evaluate the scaling capability of AREA-MAC protocol, we increase the number of nodes from 16 to 36 and check how gracefully the protocol adapts itself with the increasing number of nodes and traffic in this section. In the next section, we further increase this number to 64. For this medium size network, we keep the simulation configuration of Section 7.2.1 same for AREA-MAC nodes and evaluate their performance for energy, packet delay, data packet delivery, preambles transmission, and overheard packets metrics for both grid and random topologies. Nodes are deployed more or less in the same way as depicted in Figure 7.4 except that the grid has now a 6x6 order, and for the random topology, the sink node has been moved slightly towards the center of the playground.

7.2.5.1 Energy Consumption

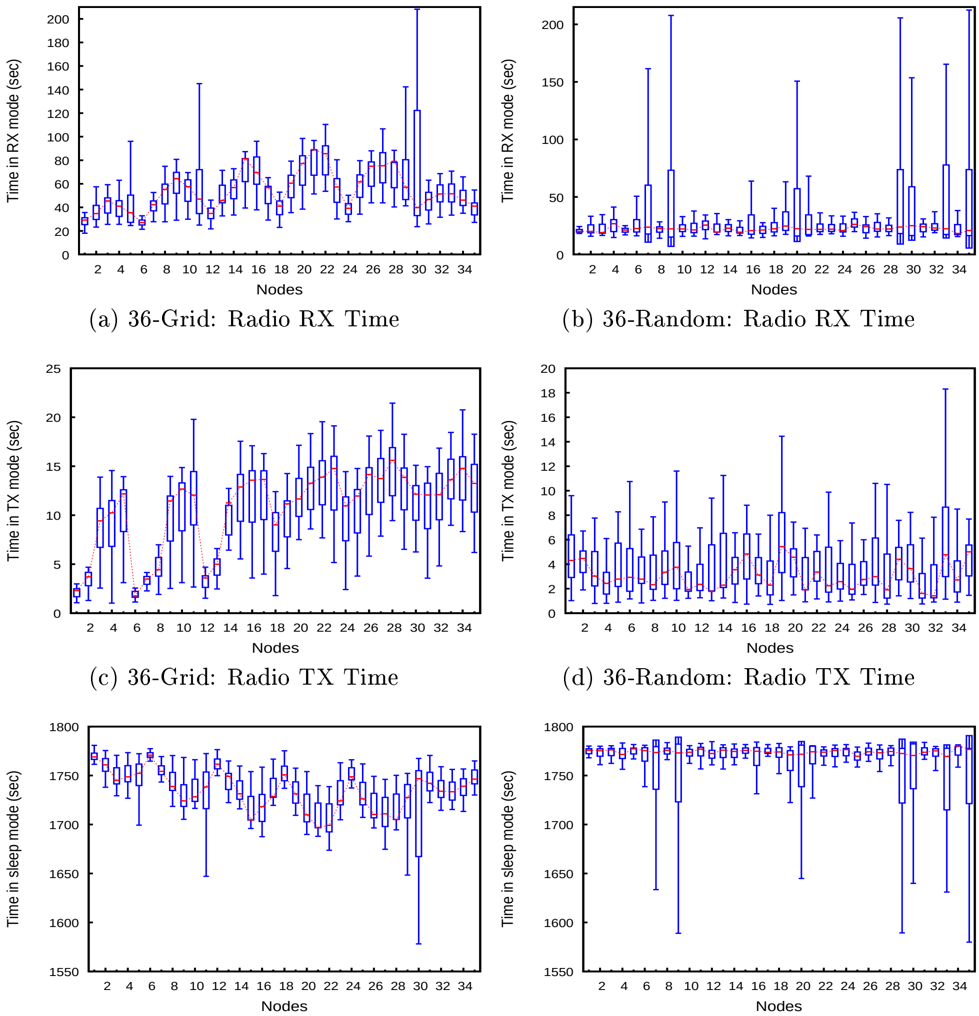
Figure 7.25 depicts the energy related performance of nodes for both AREA-MAC grid and random topologies. Again, we do not include the sink node in this analysis; the remaining 35 nodes (nodes with IDs 1 to 35) have been shown in this figure.

The radio receive time for nodes of both topologies is calculated in Figures 7.25a and 7.25b respectively. Nodes under the grid topology deplete relatively more energy in receive mode than the smaller (or normal) network scenario that is shown in Figure 7.5a. Since the number of nodes increases so does the number of data packets and preambles on the network, hence nodes receive more data packets and preambles, and their radios remain in receive mode for a longer time. Nodes that are closer to the sink node, i.e., nodes IDs with 1 and 6 receive less preambles and data packets, and they consume less energy in receive mode. As expected, the radio time in receive mode for nodes increases with their increasing distance (the number of hops) from the sink node.

For the random topology, the radio receive time of nodes remains, on average, same (around 25 seconds) for all nodes. This time is also more or less equal to the radio receive time of nodes working under the AREA-MAC random topology for the normal scenario (Figure 7.5b). Though there are more data packets and preambles transmitted over the channel for this larger network, the sink node has more one-hop neighbors (due to the slight location arrangement) than the normal scenario, and hence the radio receive time of nodes of the AREA-MAC random topology do not get affected by the increasing size of the network.

The radio transmit time of nodes for the grid topology, shown in Figure 7.25c, is higher than that of the normal scenario (Figure 7.5d) due to the increased traffic on the channel. As anticipated, the longer the distance of a node from the sink node, the higher is its radio transmit time. Nodes under the random topology, however, consume a little less energy in transmit mode (Figure 7.25d) than the normal scenario (Figure 7.5e) due to the fact that more nodes are now at the single-hop distance from the sink node.

All these points are also valid for the radio sleep time that is presented in



(a) 36-Grid: Radio RX Time

(b) 36-Random: Radio RX Time

(c) 36-Grid: Radio TX Time

(d) 36-Random: Radio TX Time

(e) 36-Grid: Radio Sleep Time

(f) 36-Random: Radio Sleep Time

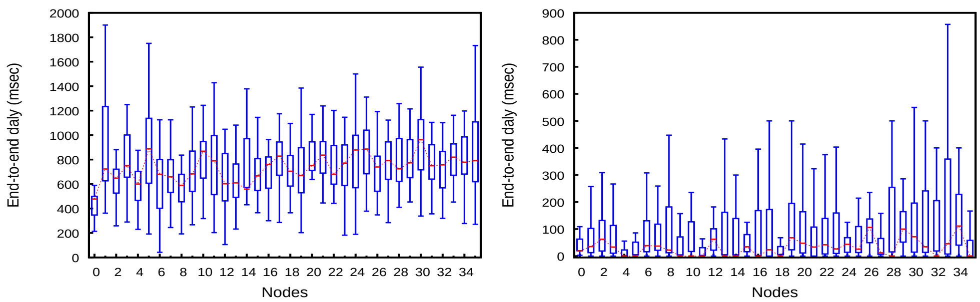
Figure 7.25: An energy consumption analysis of AREA-MAC nodes for a medium size network containing 36 nodes. (a) and (b) depict the radio receive time, (c) and (d) depict the radio transmit time, and (e) and (f) depict the radio sleep time for nodes of the grid and random topologies of the medium size AREA-MAC network respectively.

Figures 7.25e and 7.25f for nodes of AREA-MAC grid and random topologies respectively. Nodes of the grid topology consume relatively less time in sleep mode than both the random topology of this medium size network and the grid topology of the smaller network (Figure 7.5h). The AREA-MAC random topology offers more or less an equal amount of time in sleep mode for all nodes of the medium size network.

Hence, the slight movement in the location of the sink node results in better sleep time for the random topology, which is slightly higher for nodes of the grid topology. On a whole, AREA-MAC protocol provides an acceptable level of duty cycle range for nodes even with a larger network, which in fact is still better than the smaller size B-MAC network that is depicted in Figure 7.5i.

7.2.5.2 End-to-End Delay

The average end-to-end delay of the received data packets for nodes of both AREA-MAC topologies of the medium size network is depicted in Figure 7.26. The delay for the grid topology in Figure 7.26a is relatively higher than the delay of the grid topology for the smaller network shown in Figure 7.6a. This is due to the distinction that nodes by the virtue of a larger network receive and transmit more data packets, and packets generated by border nodes need to cover various multi-hop paths to arrive at the sink node.



(a) 36-Grid: E2E Delay (b) 36-Random: E2E Delay

Figure 7.26: A comparison of the end-to-end packet delay between the nodes of the (a) AREA-MAC grid topology and (b) AREA-MAC random topology for the medium size network containing 36 nodes.

The higher number of nodes with the random topology also generate, receive, and transmit more data packets, which may cover multi-hop distances to arrive at the sink node. However, relatively more nodes are now at the single-hop distance from the sink node, which almost halves the packet delay of the medium size network in Figure 7.26b as compared to that of the smaller network in Figure 7.6b. The network delay of the AREA-MAC random topology is also more than 10 times better than the AREA-MAC grid topology of the medium size network.

7.2.5.3 Packet Delivery Ratio

Like the normal traffic scenario, nodes of the medium size network generate data packets with the same data generating and deviation intervals, and therefore they generate more or less same amount of data packets with both topologies, as shown in Figures 7.27a and 7.27b.

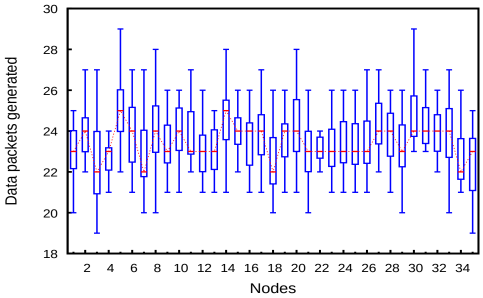
Nodes with the grid topology transmit more data packets (Figures 7.27c), especially nodes that are multi-hop away from the sink node, as compared to nodes of the normal scenario (Figure 7.7d). This is because of the larger network, where only few nodes are direct neighbors to the sink node, and other nodes keep data packets forwarding until they arrive at the sink node. Alternatively, nodes with the random topology transmit less number of data packets, as nodes usually do not need long multi-hop paths for their packets to arrive at the sink node.

The number of received data packets by nodes of both topologies is illustrated in Figures 7.27e and 7.27f. Due to the larger network, the sink node receives almost double amount of data packets than that of the normal scenario that is depicted in Figure 7.7. Grid nodes that are nearer to the sink node understandably receives less amount of packets than other nodes. Overall, nodes of the grid topology of the larger network receive a higher number of packets than nodes of the normal scenario (Figure 7.7g). Normal nodes of the AREA-MAC random topology receive, on average, less number of data packets than those of the grid topology, whereas the sink node receives more data packets (due to its better positioning) than the sink node of the grid topology.

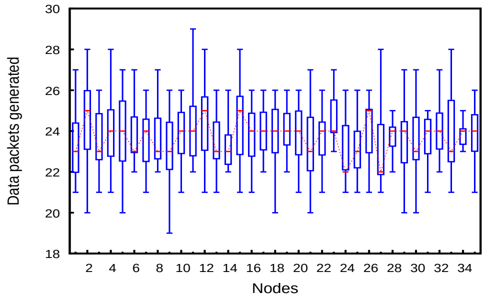
Figure 7.27g illustrates the overall packet delivery ratio at the sink node for both grid and random topologies of the medium size network. Due to the increase in the number of forwarded (multi-hop) packets, the delivery ratio of the grid topology has been decreased nearly to the 75%. It could, however, be improved if the sink node be placed at the location where few more nodes could fall into (or at least near to) its transmitting range. This is also the reason behind the improved packet delivery ratio at the sink node of the AREA-MAC random topology.

7.2.5.4 Preamble Transmission and Reception

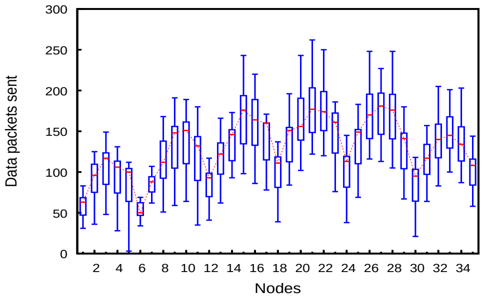
Figure 7.28 depicts the preamble transmission and reception analysis for nodes of both AREA-MAC topologies for the medium size network. Nodes of the grid topology (Figure 7.28a) transmit preambles more or less in the same way as with the normal scenario (Figure 7.8a) except that they transmit almost double amount of preambles. Since the number of nodes increases, so does the number of data packets on the channel. The sink node is kept in the left corner, which leaves border nodes to send more preambles until their duty-cycled neighbors wake up and sense the preamble. Nodes that are at the single-hop away from the sink transmit a significant lower number of preambles. Alternatively, nodes with the AREA-MAC random topology transmit less number of preambles; the actual number, however, depends on the node location.



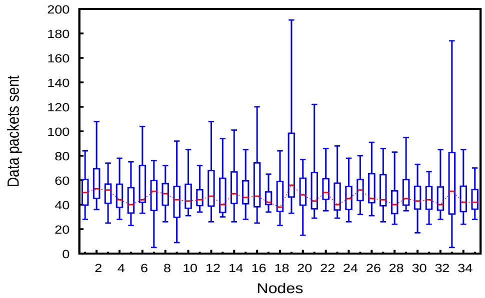
(a) 36-Grid: Generated Packets



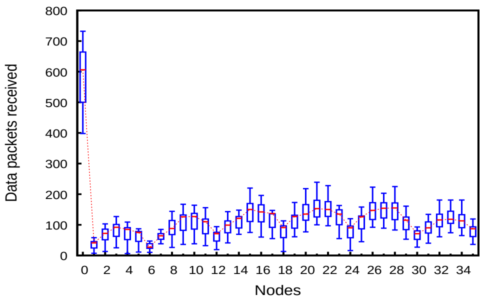
(b) 36-Random: Generated Packets



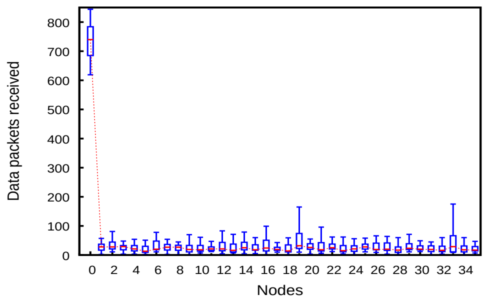
(c) 36-Grid: Sent Packets



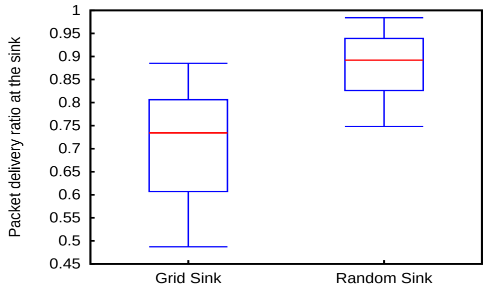
(d) 36-Random: Sent Packets



(e) 36-Grid: Recd. Packets

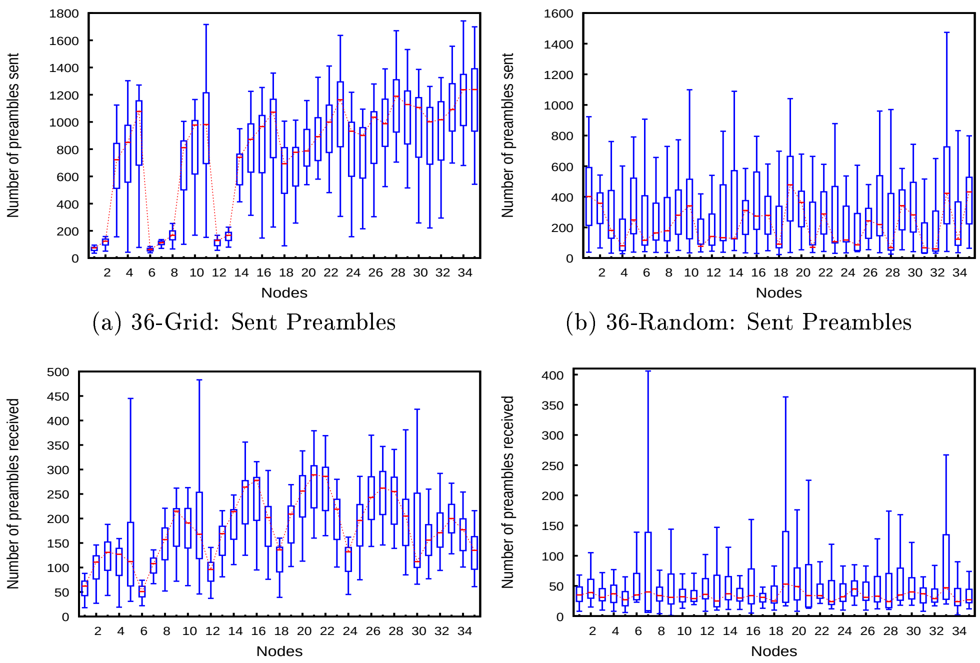


(f) 36-Random: Recd. Packets



(g) Packet Delivery Ratio

Figure 7.27: An analysis of the packet delivery ratio including data packet generation, transmission, and reception for the AREA-MAC grid and random topologies of the medium size network. (a) and (b) present the data packet generated, (c) and (d) present the data packet transmitted, and (e) and (f) present the data packet received by AREA-MAC nodes of the grid and random topologies respectively . (g) compares the overall packet delivery ratio at the sink node for both topologies for the medium size network.



(a) 36-Grid: Sent Preambles

(b) 36-Random: Sent Preambles

(c) 36-Grid: Recd. Preambles

(d) 36-Random: Recd. Preambles

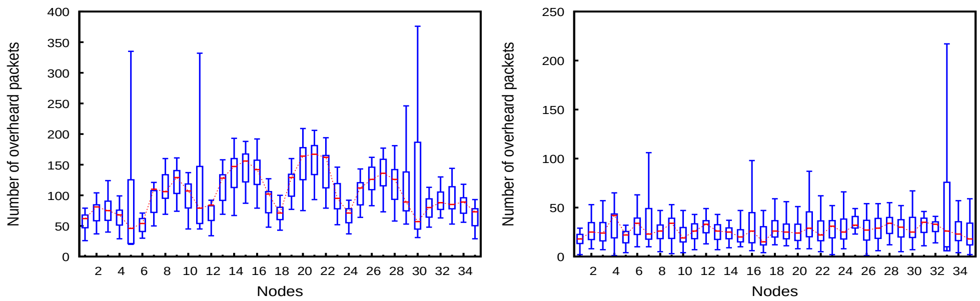
Figure 7.28: A comparison of the preamble transmission and reception for the AREA-MAC nodes of the medium size network. (a) and (b) depict the number of preambles sent and (c) and (d) depict the number of preambles received by AREA-MAC nodes of the grid and random topologies respectively.

Since nodes of the grid topology transmit a higher number of preambles, they also receive more preambles as pointed out in Figure 7.28c than the normal scenario. Nodes that are nearer to the sink node, however, receive less number of preambles. Nodes of the random topology (Figure 7.28d) receive almost an equal number of preambles.

7.2.5.5 Overheard Packets

The number of overheard packets that nodes of the medium size AREA-MAC network receive is given in Figure 7.29. Like the data packets and preambles, nodes under the AREA-MAC grid topology receive relatively a higher number of overheard packets in Figure 7.29a than nodes of the grid topology of the normal scenario (Figure 7.9a). Central nodes of the grid, i.e., nodes with IDs 7 - 9, 13 - 15, 19 - 22, and 25 - 28 receive more overheard packets than other nodes as they usually perform as forwarders for the border nodes. Nodes of the AREA-MAC random topology receive almost a constant number of overheard packets in Figure 7.29b, much in the same way as they receive with the normal scenario (Figure 7.9b).

On a whole, AREA-MAC protocol performs decently well for a larger network



(a) 36-Grid: O-heard Packets (b) 36-Random: O-heard Packets

Figure 7.29: A comparison of the overheard packets received by AREA-MAC nodes of the (a) grid topology and (b) random topology for the medium size network.

in terms of all the discussed metrics. In fact, its performance with the larger network and traffic is still better than B-MAC protocol working under the smaller network.

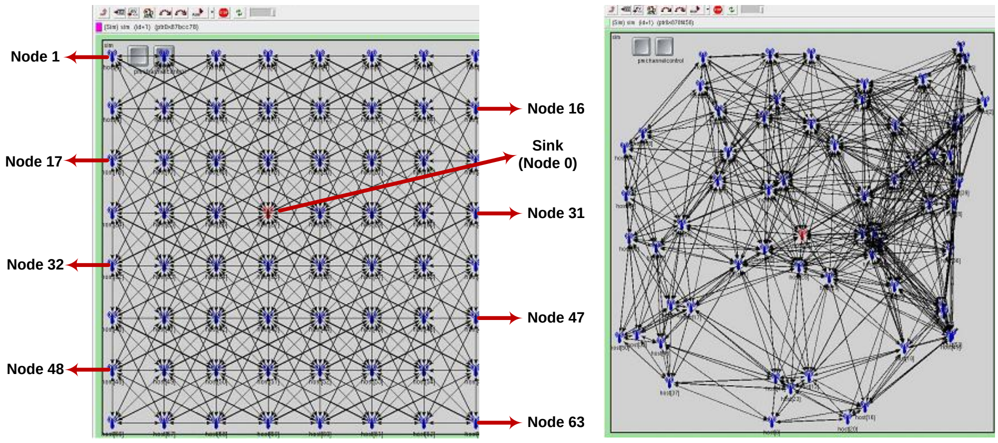
7.2.6 Scalability Effect - An Extended Network

In this section, we analyse the performance of AREA-MAC protocol for an extended network consisting of 64 nodes under both grid (with an 8x8 order) and random topologies. Figures 7.30a and 7.30b show the snapshots for both topologies. The location of the sink node has been shifted to the center of the playground for both networks. We keep the simulation configuration of Section 7.2.1 same for this scenario and evaluate the network performance for energy, packet delay, packet delivery, preamble transmission, and overheard packets metrics.

7.2.6.1 Energy Consumption

The energy related evaluation of AREA-MAC protocol for the extended network is depicted in Figure 7.31. Figures 7.31a and 7.31b establish a comparison of the radio receive time for nodes of both grid and random topologies respectively. Though the network traffic increases with the increasing number of nodes, the average radio receive time of nodes for the extended network is still less than that of the medium size network for both grid and random topologies (Figures 7.25a and 7.25b) because the sink node has been (almost) fully moved towards the center of the playground making it directly reachable to few more nodes.

Grid nodes that are equidistant from the sink node and border nodes, i.e., nodes wit IDs 14, 22, 29, and 37 of the 6^{th} column and nodes with IDs from 41 - 45 of the 6^{th} row consume more time in receive mode as they receive data packets (and preambles) from the border nodes and forward them to the sink node. Border nodes of the grid spend significantly less time in receive mode because they seldomly receive data packets from their neighbors. Alternatively, the nodes of the



(a) 64 - Grid Topology

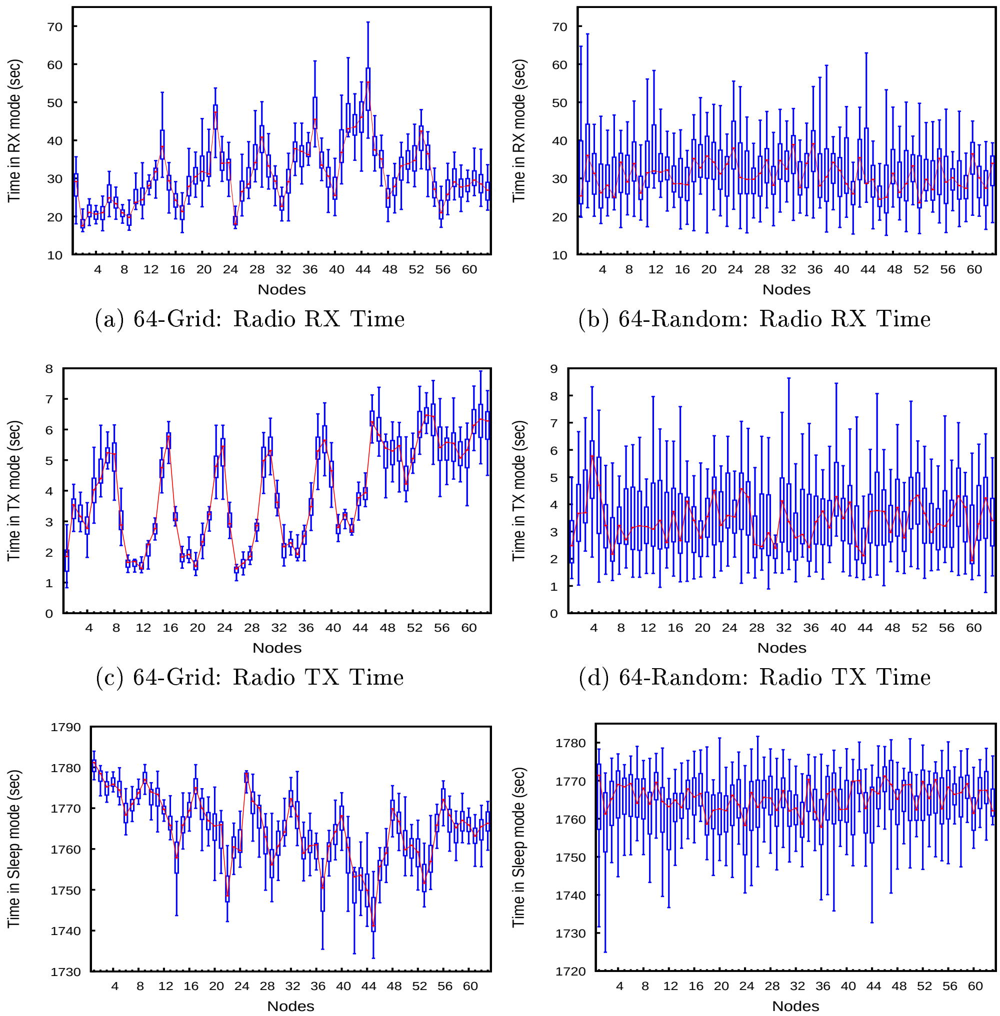
(b) 64 - Random Topology

Figure 7.30: OMNeT++ snapshots of the grid and random topologies for an extended 64-node AREA-MAC network. The red-colored node at the center represents the sink node, whereas normal nodes are the blue-colored one.

AREA-MAC random topology consume, on average, 30 seconds in receive mode for an experiment run of 1800 seconds, which is several times less than B-MAC nodes working under the smaller network with less data traffic in Figure 7.5c.

Figures 7.31c and 7.31d respectively present the radio transmit time for nodes of the AREA-MAC grid and random topologies for the extended network. Nodes, particularly the border nodes of the grid topology that are at the multi-hop distance from the sink node need to send a higher number of preambles than the single-hop nodes (of the sink node), and therefore their radios deplete almost three times more energy in transmit mode. However, the placement of the sink node at the center of the grid has reduced the overall radio transmit time almost by half as of the grid topology of the medium size network (Figure 7.25c). Alternatively, the radio transmit time of nodes of the random topology remains more or less constant and depends on the distance of the node from the sink node.

Figures 7.31e and 7.31f outline the radio sleep time for nodes of AREA-MAC grid and random topologies respectively. Since the central nodes of the grid topology consume relatively a higher time in receive mode, their sleep time is accordingly less than other grid nodes. The AREA-MAC random topology offers more or less an equal amount of time in sleep mode for all nodes. On a whole, AREA-MAC provides much lower duty cycle (on average, less than 2%) for nodes of the extended and denser network consisting of 64 sensor nodes than the traditional long preamble scheme.



(a) 64-Grid: Radio RX Time

(b) 64-Random: Radio RX Time

(c) 64-Grid: Radio TX Time

(d) 64-Random: Radio TX Time

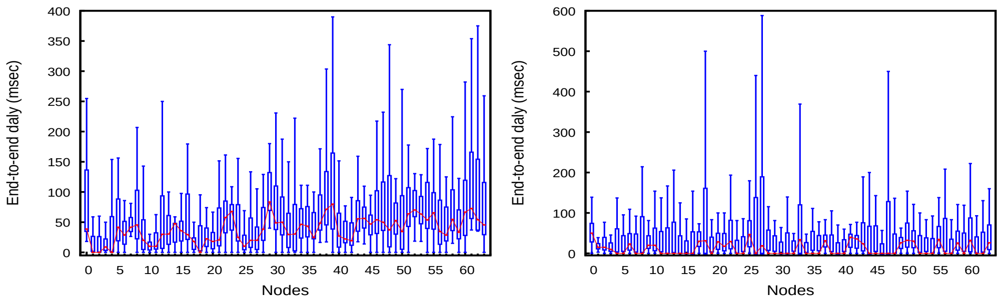
(e) 64-Grid: Radio Sleep Time

(f) 64-Random: Radio Sleep Time

Figure 7.31: An energy consumption analysis of AREA-MAC nodes for an extended network consisting of 64 nodes. (a) and (b) depict the radio receive time, (c) and (d) depict the radio transmit time, and (e) and (f) depict the radio sleep time for nodes of the grid and random topologies of the extended AREA-MAC network respectively.

7.2.6.2 End-to-End Delay

Figure 7.32 depicts the average end-to-end delay of the received data packets for nodes of both AREA-MAC grid and random topologies for the extended network. The delay at the sink node of the grid topology in Figure 7.32a is about three time less than that of the grid topology of the medium size network (Figure 7.26a) because the sink node has been almost fully displaced to the center of the grid, and therefore relatively a less number of data packets need multi-hop transmission. However, it is about three time more than the delay of the grid topology of the smaller network (Figure 7.6a) due to the increased number of nodes and data traffic on the channel.



(a) 64-Grid: E2E Delay (b) 64-Random: E2E Delay

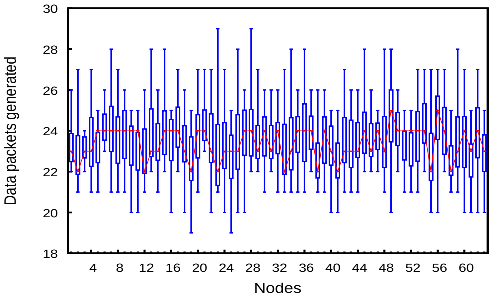
Figure 7.32: An analysis of the end-to-end packet delay between the nodes of the (a) AREA-MAC grid topology and (b) AREA-MAC random topology for the extended network consisting of 64 nodes.

Though the number of nodes and data packets increases for the AREA-MAC random topology of the extended network, the central location of the sink node nearly offsets the packet delay difference between both extended and medium size networks. Hence, the packet delay at the sink node of the random topology of the extended network in Figure 7.32b is almost equal to that of the medium size network that is depicted in Figure 7.26b. As anticipated, this delay is even smaller than the random topology of the smaller network (Figure 7.6b) and is way better than the random topology of the smaller B-MAC network (Figure 7.6c).

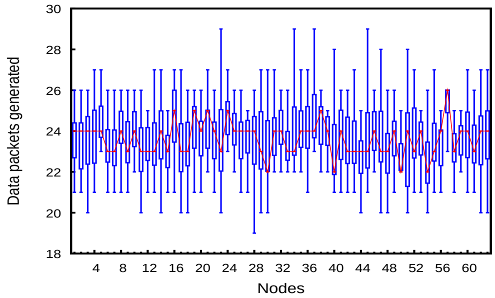
7.2.6.3 Packet Delivery Ratio

Since we have kept the data generating and deviation intervals for nodes of the extended network same as of the smaller and medium size networks, nodes of both grid and random topologies generate data packets in Figures 7.33a and 7.33b in the same fashion as with the smaller and medium size network in Figures 7.7 and 7.27 respectively.

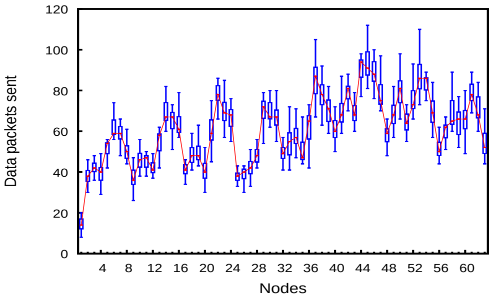
In Figure 7.33c, nodes of the AREA-MAC grid topology of the extended network transmit relatively a less number of data packets than the nodes of the grid



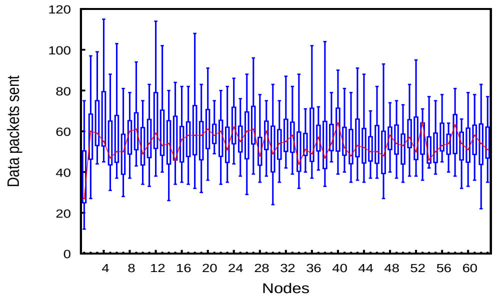
(a) 64-Grid: Generated Packets



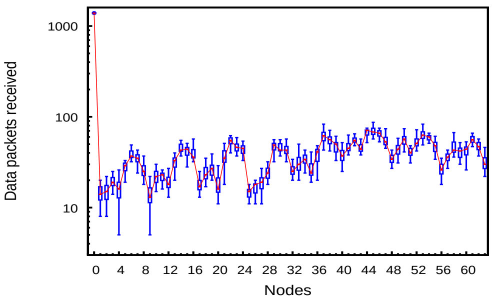
(b) 64-Random: Generated Packets



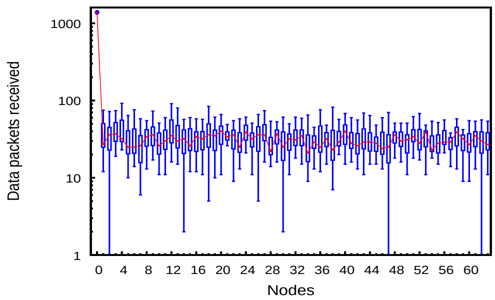
(c) 64-Grid: Sent Packets



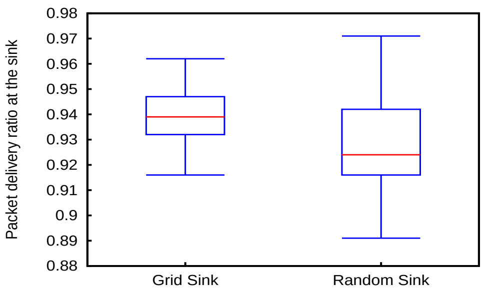
(d) 64-Random: Sent Packets



(e) 64-Grid: Recd. Packets



(f) 64-Random: Recd. Packets



(g) Packet Delivery Ratio

Figure 7.33: A comparison of the packet delivery ratio including data packet generation, transmission, and reception for the AREA-MAC grid and random topologies of the extended network. (a) and (b) present the data packet generated, (c) and (d) present the data packet transmitted, and (e) and (f) present the data packet received by AREA-MAC nodes of the grid and random topologies respectively . (g) compares the overall packet delivery ratio at the sink node for both topologies of the extended 64-node network.

topology of the medium size network (Figure 7.27c) because the sink node being at the center of the playground captures most of the data packets. However, nodes that act as forwarding nodes for the border nodes rightly send out more packets than the nodes that are the direct neighbors of the sink node. Nodes of the random topology, on the other hand, transmit more or less a constant number of data packets as per their distances from the sink node, as depicted in Figure 7.33d.

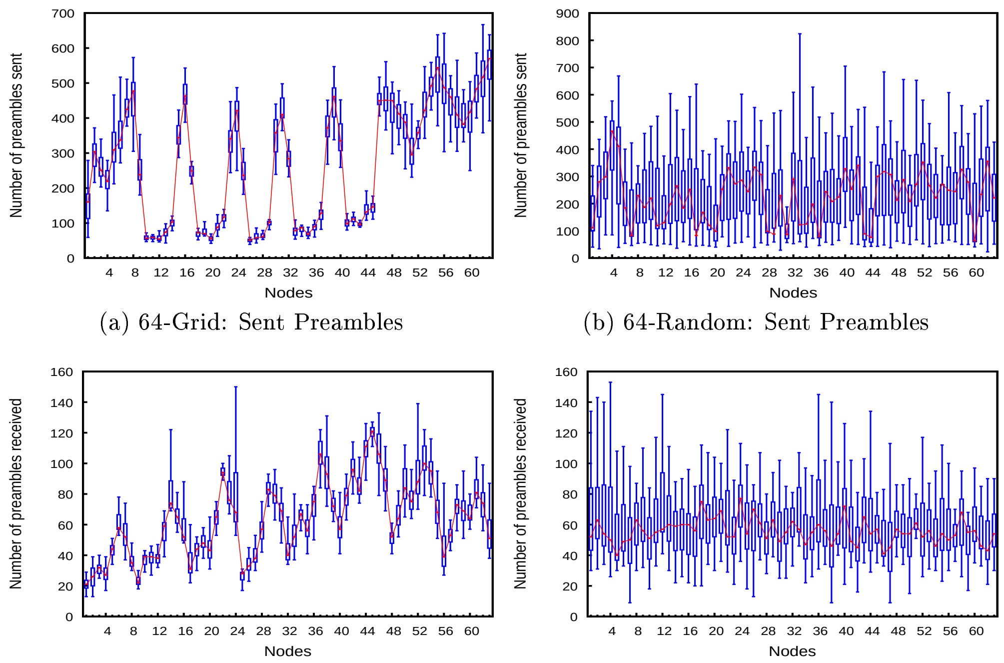
Figures 7.33e and 7.33f evident that the sink node of both topologies of the extended network receive a much higher number of data packets than the smaller and medium size networks. Nevertheless, the normal nodes of both topologies, especially the grid one, receive relatively less data packets than other networks due to the shifting of the sink location. Therefore, the overall packet delivery ratio at the sink node for both topologies in Figure 7.33 has remained above 90%, which is a very satisfactory figure for a larger and dense WSN, particularly when it is achieved with the minimal amount of energy consumption.

7.2.6.4 Preamble Transmission and Reception

The number of preambles transmitted and received by AREA-MAC nodes under both grid and random topologies of the extended network is depicted in Figure 7.34. Grid nodes transmit preambles almost in the same fashion in Figure 7.34a as with the smaller and medium size networks in Figures 7.8a and 7.28a respectively, i.e., nodes that are at the single-hop distance from the sink node transmit much less number of preambles than the multi-hop nodes. However, due to the central location of the sink node, the normal grid nodes transmit less number of preambles with the extended network than with the medium size network. Similarly, nodes of the random topology of the extended network transmit relatively less number of preambles in Figure 7.34a than the medium size network (Figure 7.28a) due to the better positioning of the sink node.

As observed in Section 7.2.6.1, the grid nodes that are almost equidistant from the sink node and other border nodes, particularly nodes of the 6^{th} column and 6^{th} row consume more energy as they receive more preambles in Figure 7.34c than other nodes due to the reason that they behave as the forwarding nodes for other border nodes. On a whole, the grid nodes of the extended network receive less number of preambles than that of the medium size network that is shown in Figure 7.28c. Alternatively, preambles received by nodes of the random topology of the extended network in Figure 7.34d is slightly higher than that of the medium size network presented in Figure 7.28d due to the transmission of a higher number of data packets on the channel. Again, the variation in the number of preambles received by nodes of the random topology depends on their distance from the sink node.

Thus, AREA-MAC nodes transmit and receive much less number of preambles for the extended network with more data traffic on the channel than the smaller network working under the long preamble scheme, which is presented in Figure 7.8.



(a) 64-Grid: Sent Preambles

(b) 64-Random: Sent Preambles

(c) 64-Grid: Recd. Preambles

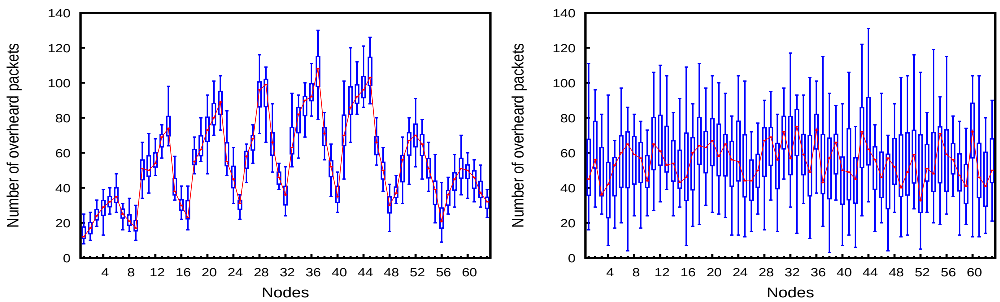
(d) 64-Random: Recd. Preambles

Figure 7.34: A comparison of the preamble transmission and reception for the AREA-MAC nodes of the extended network consisting of 64 nodes. (a) and (b) depict the number of preambles sent and (c) and (d) depict the number of preambles received by AREA-MAC nodes of the grid and random topologies respectively.

7.2.6.5 Overheard Packets

Figure 7.35 depicts the number of overheard packets received by AREA-MAC nodes of the extended network. As shown in Figure 7.35a, border nodes of the grid topology receive much less number of overheard packets than the non-border nodes, which behave as forwards for the border nodes. Nevertheless, nodes under the grid topology of the extended network receive up to 50% less overheard packets than that of the medium-size network (presented in Figure 7.29a) due to the central location of the sink node.

On the other hand, nodes of the AREA-MAC random topology of the extended network receive almost a constant number of overheard packets in Figure 7.35b, much in the same way as with the medium size network (Figure 7.29b) and the smaller size network (Figure 7.9b). The average number of overheard packets for the extended network though is relatively higher due to the higher data traffic on the channel. It is still several times less than the one that has been achieved for the smaller network working under the long preamble scheme in Figure 7.9c.



(a) 64-Grid: O-heard Packets (b) 64-Random: O-heard Packets

Figure 7.35: A comparison of the overheard packets received by AREA-MAC nodes of the (a) grid topology and (b) random topology for the extended network consisting of 64 nodes.

7.3 Discussion and Conclusion

Throughout the chapter, we have presented a detailed evaluation of AREA-MAC protocol based on its simulation implementation. Starting right from the implementation setup and configuration, several AREA-MAC results for various metrics such as energy, packet delay, packet delivery (including packet generation, transmission, and reception), preamble transmission and reception, and overheard packets have been demonstrated. AREA-MAC performs better for all the mentioned metrics and outperforms the well-known channel polling based B-MAC protocol in all the considered scenarios. AREA-MAC protocol has also been assessed for varying topology, traffic, routing, and scaling scenarios, and several results demonstrate that the protocol performs quite decently under varying traffic, scaling, and routing conditions.

AREA-MAC has initially been evaluated for a simple traffic scenario with a smaller network for both grid and random topologies. Results for the mentioned metrics have been compared with B-MAC protocol, which conclude that AREA-MAC performs much better, particularly for energy, delay, and packet delivery metrics than B-MAC protocol. Replacing long preambles with short ones and adding more information to them significantly reduce energy consumption and packet delay for nodes. One may argue that the extra information appended with every preamble would increase the processing power of a node. Since communication energy constitutes a major portion of overall energy consumption of a node (discussed in Section 2.4.2), we do not consider the processing power in our energy analysis, however the end-to-end packet delay counts the processing time of packets.

Next, the data traffic on the channel has been increased in two separate scenarios to assert the adaptability of AREA-MAC to the varying traffic conditions. In the first scenario called the burst traffic scenario, the data burst size value of nodes have been doubled, that means the number of data packets generated by nodes has also been doubled as compared to the normal traffic scenario. In the second scenario called the high traffic scenario, the data generating and deviation intervals have been decreased three times as of the simple traffic scenario that has increased the data traffic on the channel by three times. AREA-MAC performs graciously with all traffic conditions, though energy consumption and packet delays have been slightly increased. However, the adaptive duty cycling and short sleep features of AREA-MAC assist nodes to keep energy consumption and packet delay at the reasonable level, which is still much better than that of B-MAC nodes working under the normal traffic scenario. Note that the change in the location of the sink node has also assisted in the improved evaluation quality of the network.

In order to appraise the trade-off between energy consumption and packet delay, AREA-MAC protocol has then been evaluated for three different routing schemes under the grid topology. Several results comparing N_0, N_1, and N_2 schemes have been discussed, which show that though energy consumption of nodes slightly increases and the packet delivery marginally decreases with N_1 and N_2 schemes as compared to the N_0 scheme, they provide better end-to-end packet delay for nodes. Both these schemes also reduce the processing energy/time for nodes that are deployed at the border of the grid, i.e., away from the sink node. Therefore, for delay bound applications, either N_1 or N_2 routing scheme could be used, provided the location/deployment information of AREA-MAC nodes is available on hand.

To asses the behavior of the protocol for the scalability factor, AREA-MAC has been evaluated for two different WSN scaling scenarios in the last part of the chapter. The number of nodes has been increased from 16 to 36 with the medium size network and then to 64 with the extended network. Results interpret that the protocol performs acceptably well under both grid and random topologies of the scaled network. Understandably, energy consumption and packet delay for nodes, particularly nodes that are multi-hop away from the sink node, are higher than other nodes. However, the overall performance of the extended AREA-MAC network is still much better than the performance of the smaller B-MAC network working under the long traditional channel polling scheme.

After discussing an elaborated evaluation of AREA-MAC protocol, we look out for the duty cycle effect on AREA-MAC nodes and present few more results for varying duty cycle values in the next chapter. We try to find an optimal duty cycle value of nodes via the optimization method that could provide minimum energy consumption and packet delay, provided that the packet delivery ratio remains above a threshold.

Chapter 8

Optimizing Energy and Delay With AREA-MAC Protocol

Nobody can go back and start a new beginning, but anyone can start today and make a new ending.

– Maria Robinson

AFTER discussing a detailed evaluation of AREA-MAC protocol in the previous chapter, we present few more evaluation results to analyze the effect of varying duty cycle values on the performance of AREA-MAC nodes in this chapter. Subsequently, we formulate and solve different optimization problems in order to find optimal duty cycle values for nodes that could provide minimum energy consumption and packet delay, provided that the packet delivery ratio of the network remains above a threshold.

8.1 Duty Cycle Effect

As mentioned in Section 2.4.2, duty cycling is considered a meaningful approach to cope with most of the energy related issues in WSNs by putting nodes in sleep mode most of the time. Sensor nodes can preserve more energy by lowering duty cycle values to extend WSN lifetime. However, duty cycling usually results in high latency and low packet delivery ratio for a WSN. Selecting a proper duty cycle value has a lasting effect on the overall performance of a WSN, particularly on energy consumption, packet delay, and packet delivery of nodes. Choosing a long sleep period induces significant per-hop latency, since a sending node has to wait an average of half a sleep period before the receiver can accept packets. Too short sleep phases, i.e., more frequent switching of the radio between On and Off modes also limits the benefits of duty cycling.

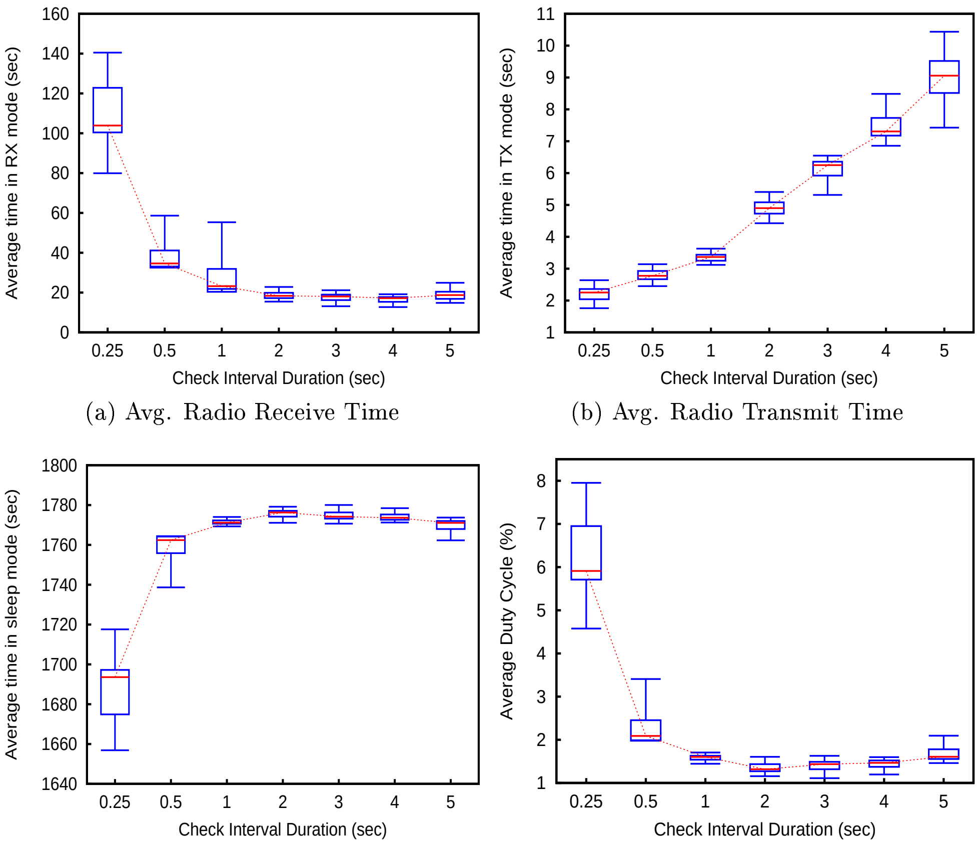
In this section, we analyze the performance of an AREA-MAC network for several duty cycle values. Since duty cycle and check interval durations of nodes are inter-related with each other (lowering the duty cycle puts nodes in sleep mode for a longer period that extends the check intervals of nodes), we consider $S = \{0.25, 0.5, 1, 2, 3, 4, 5\}$ seconds as check interval durations of nodes for this evaluation. We present results for the energy consumption, packet delay, and packet delivery ratio for AREA-MAC nodes working under the grid topology for each check interval value. The simulation configuration of the normal traffic scenario discussed in Section 7.2.1 (except for the varying check interval durations) has been used for this analysis.

8.1.1 Energy Consumption

Figure 8.1 presents an evaluation of average energy consumption of AREA-MAC nodes (expect the sink node) for different check interval values. Results have been collected for 11 different experiment runs, each lasting for 1800 seconds. Since nodes wake up quite frequently with the smaller check intervals and perform channel sampling every time they wake up, their radio receive time increases accordingly. Therefore, the radio receive time of nodes is higher for the smaller check intervals in Figure 8.1a and it plausibly decreases as the check interval increases. However, the opposite is true for the radio transmit time that increase with the increasing check interval duration in Figure 8.1b as nodes need to send out a burst of short preambles for a longer period until the respective neighbor wakes up and acknowledges.

In Figures 8.1c and 8.1d, we depict average radio sleep time and average duty cycle values of AREA-MAC nodes respectively. Both figures clearly demonstrate that nodes cannot reduce or extend their check interval durations indefinitely. Too small check interval duration (0.25 second in this case) causes for frequent radio switching between on and off modes, which consumes more node energy. Too large check interval duration forces a sending node to send out many preambles until their duty cycled neigbhors wake up and respond. Since receiving nodes wake up less frequent with the higher check interval durations, the radio transmission energy of sending nodes increases with the increasing check intervals.

For AREA-MAC protocol, the check interval duration between 1 and 3 seconds offer better energy savings for nodes, i.e., they provide, on average, less than 1.5% duty cycle. The duty cycle value of nodes increases rather dramatically as the check intervals goes below 1. It also starts increasing as the check interval is increased beyond the value of two.



(a) Avg. Radio Receive Time

(b) Avg. Radio Transmit Time

(c) Avg. Radio Sleep Time

(d) Avg. Duty Cycle

Figure 8.1: An analysis of energy consumption of AREA-MAC nodes for varying duty cycle values for 11 experiment replications, each lasting for 1800 seconds. (a) depicts the average radio receive time, (b) depicts the average radio transmit time, (c) depicts the average radio sleep time, and (d) depicts the average duty cycle of AREA-MAC nodes for different check interval values.

8.1.2 End-to-End Delay

In order to analyze the duty cycle effect on packet delay, we compare average end-to-end packet delay at the AREA-MAC sink node for various check interval values in Figure 8.2. Lower check intervals justifiably cause for lower end-to-end packet delay for nodes as neighboring nodes wake up rather frequently and acknowledge the transmitting nodes. The delay increases more or less exponentially with the increasing check interval durations. The end-to-end delay at the sink node has also to do with the number of data packets that successfully arrive at the sink node. For example, the minimum check interval value of 0.25 second results in least packet delay for the sink node, it, however, has also the least packet delivery ratio as compared to other check interval values (Section 8.1.3). Hence, a balanced check interval value for an application is usually required that could provide an acceptable packet delay as well as packet delivery ratio for a WSN.

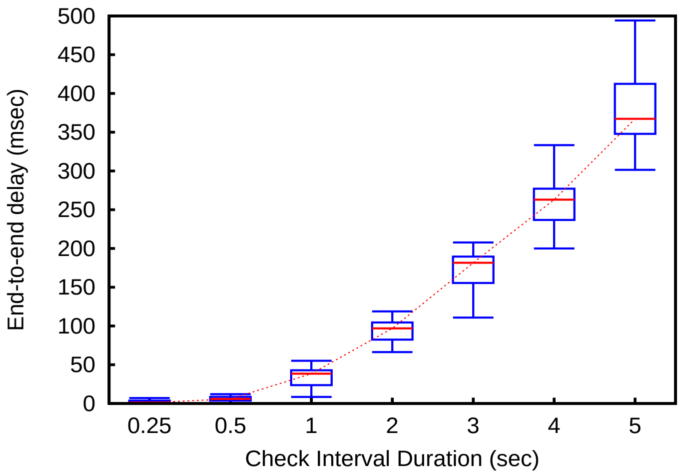
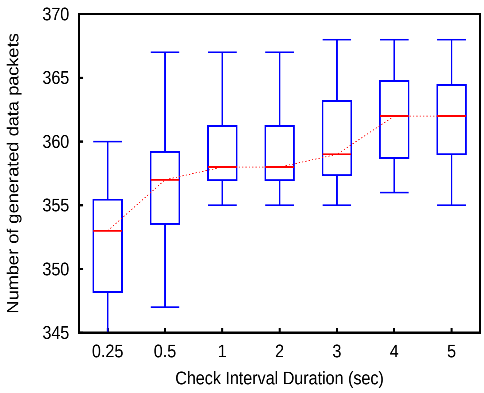


Figure 8.2: An analysis of the end-to-end packet delay at the AREA-MAC sink node for different check interval values for 11 experiment replications, each running for 1800 seconds.

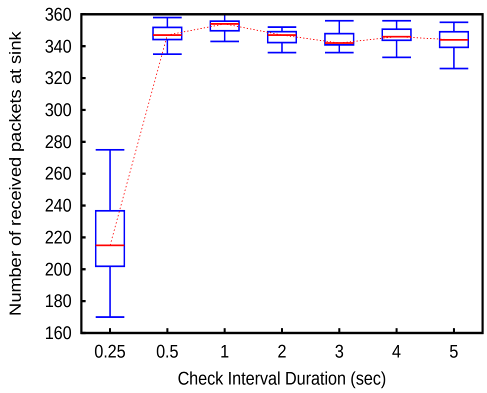
8.1.3 Packet Delivery Ratio

Figure 8.3 presents an assessment related to the data packet delivery of the AREA-MAC network for the different check interval values. The average number of data packets generated by the network is shown in Figure 8.3a. Since nodes for all replications generate data packets with the same data generating and deviation intervals for each check interval value, their average at each check interval values remains more or less constant.

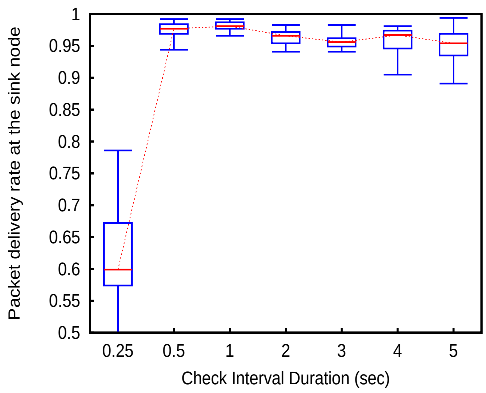
The number of data packets received at the sink node for each check interval value is calculated in Figure 8.3b, whereas Figure 8.3c presents the respective packet delivery ratio at the sink node. Like the energy consumption, the packet



(a) Generated Packets



(b) Received Packets



(c) Packet Delivery Ratio

Figure 8.3: A packet delivery analysis of AREA-MAC protocol for different duty cycle values. (a) presents the average data packet generated by AREA-MAC nodes, (b) presents the average data packet received by the AREA-MAC sink node, and (c) presents the average packet delivery ratio at the AREA-MAC sink node respectively for different check interval durations.

delivery ratio of the network also gets affected with the varying check interval durations. This figure suggests that we can not indefinitely decrease or increase the check interval durations of nodes as for as the packet delivery ratio is concerned. Very short check interval durations ($<$ 0.5 sec) significantly reduce the packet delivery ratio at the sink node because 1) nodes perform carrier sense for a constant period at each wake up duration (or at every check interval) and 2) nodes switch their radio to receive mode after sending each preamble. Both these factors leave very short time for the sending node to send preambles, particularly for the smaller check interval durations.

Similarly, as the check interval duration increases (beyond 2 sec), the wake up frequency of nodes also decreases, which means that the sending node has to send a high number of preambles. In other words, the number of preambles increases on the channel and once the (receiving) nodes are awake, they sense the carrier busy and receive the preamble and/or the subsequent data packet. Hence, their own generated data packets get delayed and the receiving queue size, particularly of the forwarding nodes keeps growing. This also decreases the packet delivery ratio of the network. On a whole, the check interval durations of 0.5 sec and 1 sec provide better (roughly 100%) packet delivery for the AREA-MAC network.

8.2 Optimizing Energy and Delay

Results discussed in the previous section demonstrate that the duty cycle value of nodes critically affects the overall performance of a WSN. Selecting an optimal check interval value for any application is not only important but also challenging, particularly when nodes generate data packets and send preambles randomly in an asynchronous network. In order to yield an optimal check interval value for a network so that energy consumption and end-to-end packet delay of a WSN can be minimized, we formulate three different optimization problems in this section. These optimization problems are related to maximizing the system lifetime, minimizing energy consumption, and minimizing end-to-end packet delay respectively.

8.2.1 Formulating Optimization Problems

Before formulating the optimization problems, we first develop formulas to calculate the lifetime of a WSN. As mentioned in Section 4.2.1, there exist several lifetime definitions for a WSN. In Section 6.2.6.1, we have defined a general and basic definition of WSN lifetime that targets at reducing the energy consumption of nodes. A more specific definition of the WSN lifetime for AREA-MAC protocol is given below [127].

WSN Lifetime: If we redraw Figure 6.1 to add durations for preamble, pre-ACK, and data packets, we get Figure 8.4. After sending a short preamble, the sending nodes waits in receive mode to acquire a pre-ACK from the receiver. For

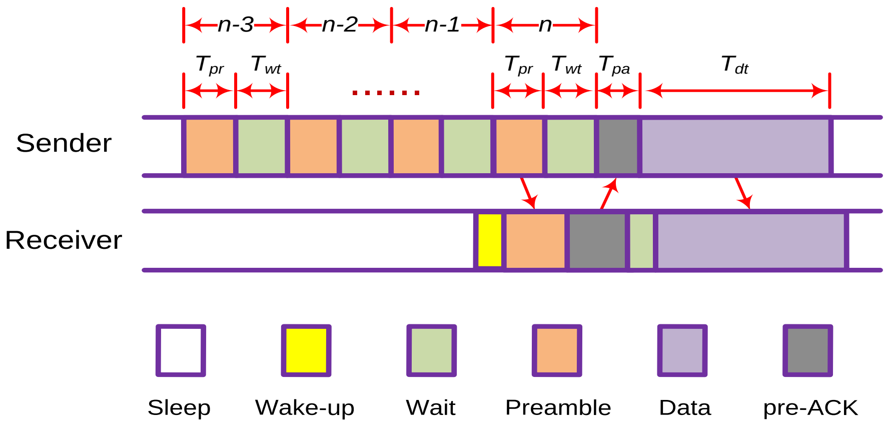


Figure 8.4: The basic working of AREA-MAC protocol including durations for sleep, wake-up, pre-ACK, and data packets. T_{pr} *is the preamble transmission/reception period,* T_{wt} *is the waiting period,* T_{pa} *is the pre-ACK transmission/reception period, and* T_{dt} *is the data transmission/reception period. Both* T_{pr} *and* T_{wt} *have been combined into an* n*.*

simplicity, we add the preamble period T_{pr} and waiting period T_{wt} into one period n. The sending node keeps sending short preambles for a complete check interval duration until the intended receiver wakes up and acknowledges the sender. If the receiver does not hear first $(n-1)$ preambles, then the sending node v_i sends out the n-th preamble in the following time interval.

$$\left[\left(\left(T_{pr}+T_{wt}\right)(n-1)-T_{pr}\right),\ \left(\left(T_{pr}+T_{wt}\right)n-T_{wt}\right)\right]$$

For a neighboring node v_j to wake up and hear the n-th preamble - provided that it wakes up only for a short period to sense the existence of the preamble - it needs to wake up during the following interval.

$$\left[\left((T_{pr}+T_{wt})(n-1)-T_{pr}-(3/4)\,T_{wt}\right),\ \left(\left(T_{pr}+T_{wt}\right)n-T_{wt}-(1/4)\,T_{pr}\right)\right]$$

Lets consider that the check interval duration of a node is determined by the Poisson process with the rate of λ with which each node v_j wakes up. The probability p that the neighboring node v_j wakes up and receives the n-th preamble sent by v_i can then be given as:

$$p = 1 - e^{-\lambda(T_{pr}+T_{wt})}$$

If we assume that ε is the initial energy available to the node v_i that consumes α unit of energy each time it wakes up, then the expected lifetime of the node v_i is given by:

$$E\left(L^{v_i}\right) = \frac{\varepsilon}{\alpha\lambda}$$

For a given awake probability p, the lifetime of the node v_i can be expressed as follows:

$$\begin{aligned} L_p^{v_i} &= \frac{1}{\beta\lambda} \\ &= \frac{T_{pr} + T_{wt}}{\beta\, ln\frac{1}{1-p}} + T_{dt} \end{aligned} \tag{8.1}$$

Where $\beta = \alpha/\varepsilon$ is the power consumption ratio of the node v_i.

8.2.1.1 Maximizing Network Lifetime

In the first optimization problem (OP-1), we aim to maximize the overall network lifetime of an AREA-MAC network. The objective function (8.2a) maximizes the lifetime of an AREA-MAC node v_i with respect to the given awake probability p, and by keeping delay and packet delivery ratio as per their thresholds. Constraint (8.2b) limits end-to-end packet delay D_{v_i,v_0} below the threshold τ. Constraints (8.2c) and (8.2d) keep packet delivery ratio pdr above the threshold ϑ and below 1 respectively. Non-negativity constraints in (8.2e) prevent values going below 0.

OP-1

$$\max_{p} \quad L_p^{v_i} \qquad v_i \in V;\ i > 0 \tag{8.2a}$$

subject to:

$$D_{v_i,v_0} \leq \tau \qquad v_i, v_0 \in V;\ i > 0 \tag{8.2b}$$

$$pdr \geq \vartheta \tag{8.2c}$$

$$pdr \leq 1 \tag{8.2d}$$

$$\big[p,\ D_{v_i,v_0},\ \tau,\ \vartheta\big] \geq 0 \tag{8.2e}$$

8.2.1.2 Minimizing Energy Consumption

The second optimization problem (OP-2) minimizes the wake up time of node v_i for the given value of check interval duration in (8.3a). As the minimum wake up time of a node decreases its radio energy consumption, which ultimately decreases the duty cycle of the node and increases the overall WSN lifetime. Constraint (8.3b) limits end-to-end packet delay D_{v_i,v_0} below the threshold τ. Constraints (8.3c) and (8.3d) keep pdr above the threshold ϑ and below 1 respectively. Constraints in (8.3e) are the non-negativity constraints.

OP-2

$$\min_{ci} \quad T_{wk}^{v_i} \qquad v_i \in V;\ i > 0 \qquad (8.3a)$$

subject to:

$$D_{v_i,v_0} \leq \Omega \qquad v_i, v_0 \in V;\ i > 0 \qquad (8.3b)$$

$$pdr \geq \vartheta \qquad (8.3c)$$

$$pdr \leq 1 \qquad (8.3d)$$

$$\left[ci,\ D_{v_i,v_0},\ \Omega,\ \vartheta\right] \geq 0 \qquad (8.3e)$$

8.2.1.3 Minimizing End-to-End Delay

The last optimization problem (OP-3) minimizes the end-to-end delay at the sink node for packets that are generated and sent by node v_i in (8.4a). For this optimization problem, energy consumption and packet delivery ratio of the AREA-MAC network have been considered as the constraints. Constraint (8.4b) limits the wake up time of node v_i below the threshold ω. Constraints (8.4c) and (8.4d) keep *pdr* above the threshold ϑ and below 1 respectively. Constraints in (8.4e) are the non-negativity constraints.

OP-3

$$\min_{ci} \quad D_{v_i,v_0} \qquad v_i, v_0 \in V;\ i > 0 \qquad (8.4a)$$

subject to:

$$T_{wk}^{v_i} \leq \omega \qquad v_i \in V;\ i > 0 \qquad (8.4b)$$

$$pdr \geq \vartheta \qquad (8.4c)$$

$$pdr \leq 1 \qquad (8.4d)$$

$$\left[ci,\ T_{wk}^{v_i},\ \omega,\ \vartheta\right] \geq 0 \qquad (8.4e)$$

8.2.2 Drawing Relationship Between Parameters

The exact calculation for such optimization problems is a difficult task since data packets generation and preamble transmission are randomized, and sleep and wake-up time of nodes are asynchronous. Based on the results that we have obtained in the previous section, we next solve the last two optimization problems. Approximate relationships between different parameters are first drawn by applying the linear regression, and then these relationships are fine tuned empirically for both optimization problems in order to efficiently present the results.

Subsequently, we achieve the following relationships between energy, delay, and packet delivery ratio for the given value of check interval duration. Since these parameters change considerably with the changing value of check interval duration, we divide them into three different groups of check interval durations (i.e., 0.25, 0.5 - 1, and 2 - 5 seconds), so that a close relationship between them can be made.

$$energy \cong \begin{cases} 1.78ci + 113.5 & ci = 0.25 \\ -22.7ci + 51.23 & 0.5 \leq ci \leq 1 \\ 1.78ci + 20.21 & 2 \leq ci \leq 5 \end{cases} \tag{8.5}$$

$$delay \cong \begin{cases} 0.007ci & ci = 0.25 \\ 0.05ci - 0.02 & 0.5 \leq ci \leq 1 \\ 0.09ci - 0.1 & 2 \leq ci \leq 5 \end{cases} \tag{8.6}$$

$$pdr \cong \begin{cases} -0.01ci + 0.62 & ci = 0.25 \\ 0.01ci + 0.97 & 0.5 \leq ci \leq 1 \\ -0.01ci + 0.99 & 2 \leq ci \leq 5 \end{cases} \tag{8.7}$$

8.2.3 An Optimal Solution

Depending on the relationships achieved in Equations (8.5), (8.6), and (8.7), both optimization problems have been solved for the optimal values of energy and packet delay. As pointed out in Table 8.1, the check interval value of 2 second provides minimum energy consumption (in terms of wake-up time of nodes) for AREA-MAC protocol. Alternatively, the check interval duration value of 0.5 second provides the minimum end-to-end packet delay for an AREA-MAC based WSN. The packet delivery ratio for the check interval value of 0.5 second is slightly better than the value of 2 second. Maximum values for the thresholds Ω, ω, and ϑ are 4 seconds, 65 seconds, and 0.9 for delay, energy, and packet delivery ratio respectively.

Table 8.1: Optimal solution in terms of minimum energy (minimum wake-up time) of nodes and minimum end-to-end packet delay at the sink node for an AREA-MAC network

Objective Fn.	**check interval**	**wake-up time**	**delay**	**pdr**
minimize energy	2 s	23.77 s	80 ms	0.97
minimize delay	0.5 s	39.88 s	5 ms	0.98

Though the check interval duration of 0.25 seconds offers better end-to-end packet delay than the check interval of 0.5 seconds, the packet delivery ratio with 0.25 seconds is much less than the required ϑ threshold. Therefore, the check interval value of 0.5 seconds provides optimal packet delay with an acceptable packet delivery ratio.

8.3 Conclusions

In this chapter, we have studied the effect of varying duty cycle values on the energy consumption, end-to-end packet delay, and packet delivery ratio of AREA-MAC nodes. Through results, we have analyzed that nodes cannot indefinitely reduce or extend their duty cycle values. Choosing long sleep periods results in significant per-hop delay and higher transmission energy, since a sending node has to send many preambles until neighboring nodes wake up and acknowledge. Alternatively, too short sleep phases increase the radio receive time of nodes as they wake up and sense the channel quite frequently. Hence, there exists an optimal duty cycle value for nodes beyond that their performance (in terms of energy consumption, packet delay, and packet delivery) drops-off.

To find such an optimal duty cycle value for AREA-MAC nodes, we subsequently have formulated and solved different optimization problems against a wide range of check interval durations for several experiment replications, each lasting for 1800 seconds. Nodes achieve optimal (minimum) energy consumption with the least wake-up time of 23.77 seconds at the check interval value of 2 seconds. Whereas, the optimal (minimum) end-to-end packet delay of 5 ms has been observed at the check interval value of 0.5 seconds, which also offers better packet delivery ratio for the AREA-MAC network.

In the next chapter, the implementation and performance evaluation of AREA-MAC protocol on the testbed platform and its comparison with the simulation platform are discussed.

Chapter 9

Performance Evaluation of AREA-MAC: A Testbed Approach

The strongest arguments prove nothing so long as the conclusions are not verified by experience. Experimental science is the queen of sciences and the goal of all speculation.

– Roger Bacon

THIS chapter presents the real-world implementation and performance evaluation of AREA-MAC on the DES-Testbed platform. The protocol has been evaluated for several metrics which include energy consumption, packet delay, packet delivery, preamble transmission/reception, and overheard packets for AREA-MAC nodes. At the end of the chapter, an appraisal comparison between the testbed evaluation and simulation evaluation of AREA-MAC protocol will be carried out in order to look into the legitimacy of the used simulation models.

9.1 Evaluation Setup

We have implemented and evaluated AREA-MAC on the DES Testbed platform [49]. The hardware specification of the DES Testbed has been discussed in Section 4.1.3. In this section, we give a further insight into the parameter selection, packet formatting, and node deployment for the testbed implementation of AREA-MAC protocol.

9.1.1 Parameters Setting

Before presenting the selection of different parameters, we first detail the concept of Wake-On-Radio (WOR) module that is an important characteristic of CC1100 transceiver to provide duty cycle option to sensor nodes.

9.1.1.1 Wake-On-Radio (WOR):

Apart from constant receive mode, CC1100 radio chip offers a low-energy based WOR functionality that enables radio of a node to perform duty cycle, i.e., to periodically wake up from sleep mode and listen to incoming packets. After waking up, the on-chip timer sets radio into the idle and then receive mode. Radio switches back to sleep mode after a programmable time in receive mode, unless a packet is received. Once radio senses the preamble on the channel, it extends its time in receive mode. Figure 9.1 sketches the switching time and the estimated current relationship between different modes of CC1100 transceiver [128].

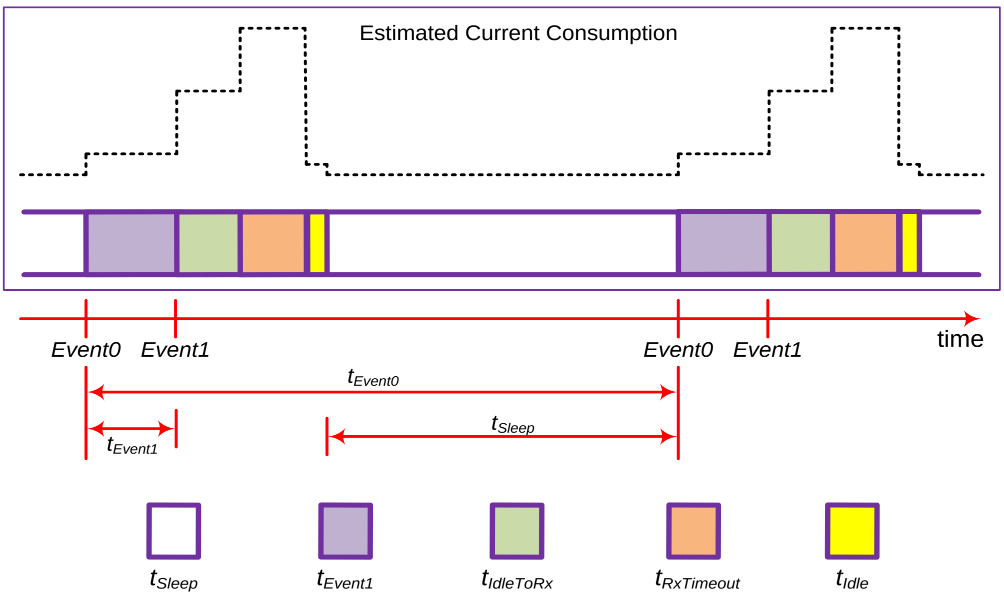


Figure 9.1: The figure shows transition times and the respective estimated current consumption (on the top) of different modes of CC1100 transceiver [128].

By putting radio in duty cycle rounds, WOR module conserves a significant amount of energy for sensor nodes. Another advantage with WOR is that the MCU stays in sleep mode while radio performs channel polling. It needs to wake up when radio actually detects a packet. The time radio wakes up from sleep mode is denoted by *Event0*, and the time it switches to receive mode is denoted by *Event1*. Reaching *Event0* turns on the digital regulator and starts the crystal oscillator (XOSC). Since XOSC powers down during sleep mode, the time between an *Event0* and the following *Event1* must be long enough for the XOSC to start

Table 9.1: Duty cycle approximation for CC1100 radio chip [128]

RX_TIME [2:0]	*WOR_RES = 0*	*WOR_RES = 1*
0 (000)	12.5%	1.95%
1 (001)	6.25%	9765 ppm
2 (010)	3.125%	4883 ppm
3 (011)	1.563%	2441 ppm
4 (100)	0.781%	NA
5 (101)	0.391%	NA
6 (110)	0.195%	NA
7 (111)	Until end of packet	Until end of packet

after the wake-up. The time between two consecutive *Event0*s (or *Event1*s) is called a wake-up interval t_{Event0} and is given by:

$$t_{Event0} = \frac{750}{f_{XOSC}} \cdot EVENT0 \cdot 2^{(5 \cdot WOR\_RES)} \tag{9.1}$$

Where f_{XOSC} is the oscillator frequency and *WOR_RES* bits of the *WORCTRL* register control the time resolution of t_{Event0}, and hence the maximum time-out of WOR module. The possible values of *WOR_RES* with the WOR setup are 0 and 1. The RX timeout of the WOR, $t_{RxTimeout}$, is the timeout for sync word search and is calculated by the *RX_TIME* field of the *MCSM2* register as per duty cycle values given in Table 9.1. The equation to calculate $t_{RxTimeout}$ is:

$$t_{RxTimeout} = \frac{t_{Event0}}{2^{(RX\_TIME+3+WOR\_RES)}} \tag{9.2}$$

Figure 9.2 depicts the WOR setting and the sending and receiving processes of AREA-MAC nodes for the DES-Testbed implementation. For this implementation, we have chosen 500 ms as the Event0 value, which means that nodes wake up at every 500 ms to sense the existence of a preamble on the channel. The sending node transmits continuous preambles up to the *burst count* number, which is nearly equal to 170 for a t_{Event0} duration of 500 ms. The gap between two consecutive preambles is equal to the $t_{packet\_interval}$ duration having value of 3 ms. The sender stops sending preambles as soon as it receives a pre-ACK from the intended receiver. Since $t_{packet\_interval}$ needs to be less than $t_{RxTimeout}$, this time corresponds to the 0.781% duty cycle for the *WOR_RES* value of 0 and *RX_TIME* value of 4 as shown in Table 9.1. This sets $t_{RxTimeout}$ value of 3.905 ms.

Once the receiving node senses a preamble, it cancels its $t_{RxTimeout}$ and extends the wake-up duration in order to receive the preamble. If the node is the correct destination the preamble is destined to, it sends a pre-ACK and then receives the following data packet from the sender. After processing, radio turns into the idle state. The MCU then needs to manually put back the radio into sleep mode from the idle state by using a *SWOR* strobe.

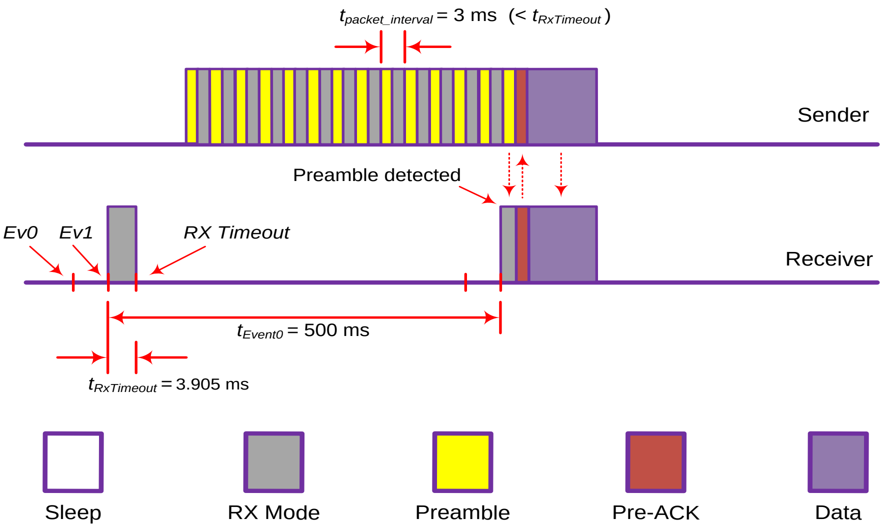


Figure 9.2: CC1100 WOR setting including the transmission and reception processes of nodes for the AREA-MAC implementation on the DES-Testbed platform.

Table 9.2 presents the overall parameter selection for the AREA-MAC implementation on the DES-Testbed platform. Most of the parameters have been selected in accordance with the simulation implementation of AREA-MAC protocol (Chapters 7 and 8), so that a general and broader relationship between both platforms can be established. The RSSI value, which is an estimate of the signal power level in the chosen channel, is calculated in dBm by the help of RSSI offset value and the current $RSSI_d$ (decimal) value of the RSSI status register as follows [110]:

$$RSSI_{dBm} = \begin{cases} (RSSI_d - 256)/2 - RSSI_{offset}, & \text{if } RSSI_d \geq 128 \\ (RSSI_d)/2 - RSSI_{offset}, & \text{if } RSSI_d < 128 \end{cases} \quad (9.3)$$

9.1.2 Packets Formatting

As mentioned in Section 7.1.2, AREA-MAC nodes use three types of frames, namely preamble, data, and pre-ACK frames to communicate with each other. Figure 9.3 presents the bit configuration of each frame type for the AREA-MAC implementation on the DES-Testbed. Note that the size of each frame is chosen in relevance with the CC1100 chip configuration. Physical preamble, sync, and CRC bits of each frame are added automatically by CC1100 transceiver during the transmitting process and removed during the receiving process. A physical preamble is used to learn and synchronize with the right phase of the sending node.

Bits 0 - 3 of the *flags* byte are used to identify the type of the frame (preamble, pre-ACK, or data). The 4^{th} bit of the *flags* byte is used to differentiate the

Table 9.2: Implementation configuration of the AREA-MAC protocol on the DES-Testbed platform.

Layer	**Parameter**	**Value**
App layer	Number of nodes	16
	Topology	Random
	Experiment runs	11
	Experiment time/run	1800 s
	Data generating interval	75 s
	Data generating deviation	90 s
	Data packet length	42 bytes
	Data burst size	1
MAC layer	Queue size	10
	Check interval (WOR Event0)	500 ms
	WOR Event1	1.3 ms
	WOR RxTimeout	3.905 ms
	WOR packet interval	3 ms
	Preamble length	14 bytes
	pre-ACK length	12 bytes
	Preamble burst count	170
	CS threshold	-91 dBm
PHY layer	Transmission power	+5 dBm
	Carrier frequency	868 MHz
	Radio channel	5
	XOSC frequency	26 MHz
	Data rate	400 Kbps
	RSSI offset	77 dB
	Modulation scheme	MSK

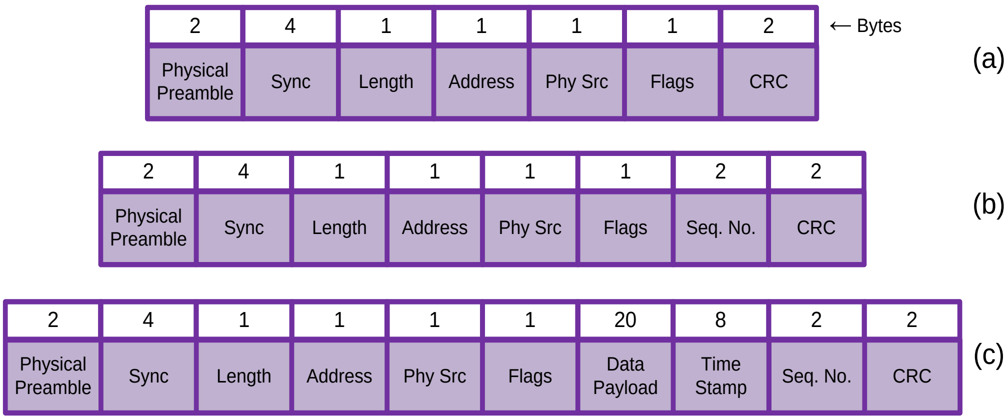


Figure 9.3: Bit configuration for the (a) AREA-MAC pre-ACK, (b) AREA-MAC preamble, and (c) AREA-MAC data packet frames, each of 12, 14, and 42 bytes long respectively.

message type (normal or real-time) for the preamble frame and as a *more-to-follow* bit for the data frame to indicate whether more data packets are waiting in the transmitting buffer to be transmitted.

9.1.3 Testbed Deployment for the AREA-MAC Implementation

Since the simple traffic scenario of the AREA-MAC simulation in Chapter 7 has been carried out on 16 nodes, we implement and evaluate AREA-MAC testbed implementation on 16 nodes. These nodes are deployed at the Institute of Computer Science, Freie Universität Berlin and are shown in Figure 9.4. The sink node (red circled Node 0 in the figure) has been chosen in accordance with the maximum node connectivity with other nodes. Few nodes are at the single-hop distance, whereas others are multi-hop away from the sink node. All nodes are not deployed in the consecutive rooms, therefore empty rooms (rooms without nodes), walls, corridor, etc., cause communication interference and force nodes to chose alternate paths.

9.2 Evaluation Results

In this section, we present and discuss various results that have been obtained from several testbed experiments to evaluate AREA-MAC protocol. As mentioned, the protocol has been assessed for energy consumption, packet delay, packet delivery (including data packet generation, transmission, reception, and delivery ratio), preamble transmission and reception, and overheard packets metrics.

9.2.1 Energy Consumption

Like the simulation analysis, we calculate energy consumption of a DES-node working under AREA-MAC protocol by counting times its radio spends in the receive,

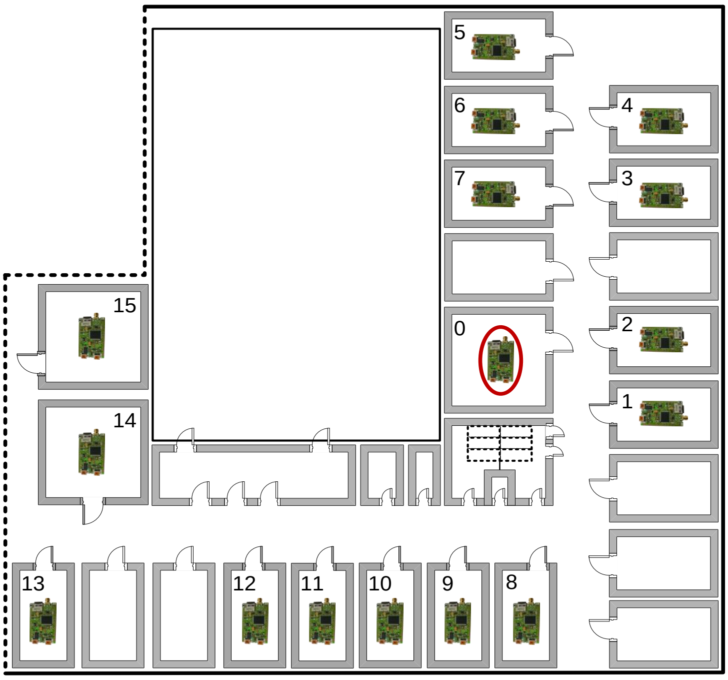


Figure 9.4: The deployment of nodes on the DES-Testbed platform at the Institute of Computer Science, Freie Universität Berlin to implement and evaluate AREA-MAC protocol. The red circled node 0 is the sink node. Other nodes generate, transmit, and forward data packets to the sink node.

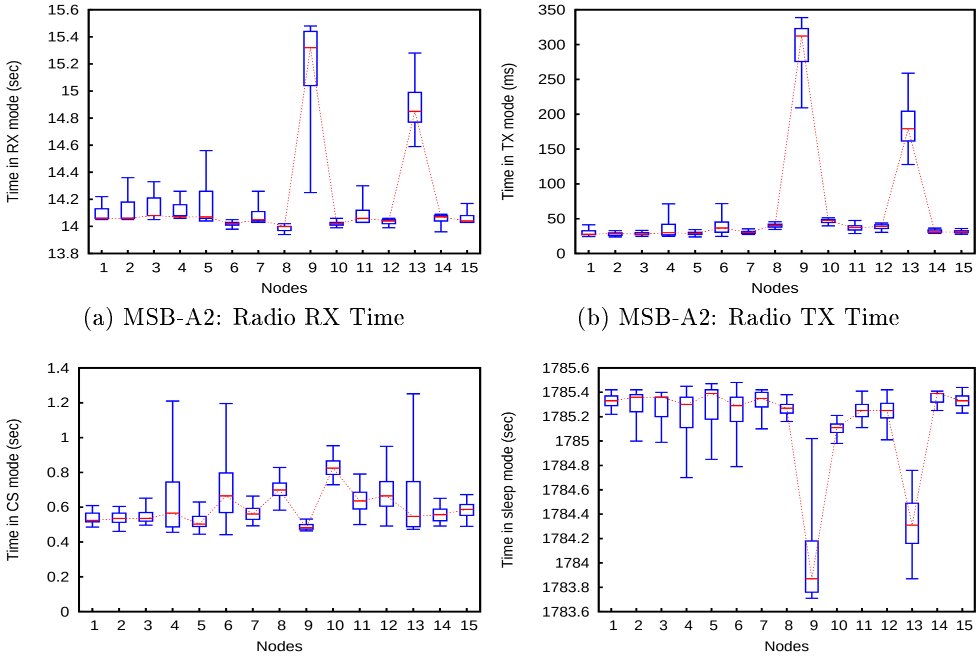
transmit, and sleep mode respectively. However, the carrier sense time of the node has been excluded from the radio receive time and is calculated separately. All the energy related results are depicted in Figure 9.5. Since the sink node remains constantly in receive mode throughout the experiments, we do not include it in the energy related evaluation.

Figure 9.5a shows the radio receive time of nodes for the AREA-MAC DES-Testbed implementation. The receive time of nodes can be seen as a sum of three times:

1. Nodes periodically wake up at every *Event0* interval (500 ms in our case) and remain in receive mode for the $t_{RxTimeout}$ of 3.905 ms, which is extended if the node receives a packet. On average, nodes receive frames (if any) at the half of the $t_{RxTimeout}$.

2. Once nodes sense the preamble, they remain in receive mode in order to receive frame(s). Note that the total number of received packets for a node also includes overheard packets that are mostly preamble ones.

3. After sending a preamble, radio of the sending node by default switches to receive mode for half of the $t_{packet\_interval}$ duration, i.e., for 1.5 ms in order to receive the pre-ACK from the intended receiver.



(a) MSB-A2: Radio RX Time

(b) MSB-A2: Radio TX Time

(c) MSB-A2: Radio CS Time

(d) MSB-A2: Radio Sleep Time

Figure 9.5: An analysis of the radio energy consumption of MSB-A2 nodes working under AREA-MAC protocol. (a) shows the radio receive time, (b) shows the radio transmit time, (c) shows the radio carrier sense time, and (d) shows the radio sleep time of DES WSN nodes.

Hence, all three times add up to the total radio receive time for a node. All AREA-MAC nodes (except nodes with IDs 9 and 13) consume more or less an equal time in the receive mode, which is about 14 seconds for an experiment run of 1800 seconds. Nodes with IDs 9 and 13 have marginally higher radio receive time mainly due to their physical locations (Figure 9.4). Node 13, which is at the multi-hop distance from the sink node, can communicate with the sink node only via node 14. Similarly node 9 is more than two-hop away from the sink node and can communicate with the sink via either 9 $\rightarrow$ 8 $\rightarrow$ 1 $\rightarrow$ 0 or 9 $\rightarrow$ 10 $\rightarrow$ 11 $\rightarrow$ 0 route. The latter route is a valid route due to the open space between nodes 11 and the sink node, where they can communicate directly with each other through glass windows/doors. In addition, node 9 seems to have some hardware issues, particularly with its antenna. Therefore, both nodes 9 and 13 need to send more preambles until their respective neighbors wake up and respond. Since nodes switch to receive mode after sending each preamble in order to get the pre-ACK from the neighbors, both these nodes consume a slightly higher time (up to 1 second) in receive mode than other nodes.

The radio transmit time of AREA-MAC nodes, which has been calculated in milli-seconds, is shown in Figure 9.5b. Nodes usually transmit a higher number of preambles than data packets and/or pre-ACKs, therefore the radio transmit time mostly depends on the number of preambles that a node transmits. Since node 9 and 13 transmit a relatively higher number of preambles than other nodes, their radio transmit times are also higher. Remaining nodes, on average, spend 40 ms in transmitting frames (preambles, data, and pre-ACK) for an experiment run of 1800 seconds.

Figure 9.5c depicts the carrier sense time of each AREA-MAC node for the DES-Testbed implementation. Since nodes 8 and 10 send out more data packets by serving as forwarding nodes for node 9, they have a little higher radio carrier sense time (up to 200 ms) than other nodes. On the whole, nodes that are at the single-hop distance from the sink node, i.e., nodes with IDs 1, 2, 3, 7, 14, and 15 consume relatively less time in sensing the carrier than nodes that are multi-hop away from the sink node as single-hop nodes could communicate with the sink node in the first attempt of preamble transmission.

The overall radio sleep time of nodes is depicted in Figure 9.5d. AREA-MAC provides ultra low duty cycle (on average less than 1%) for nodes that reflects the efficiency of the protocol for low energy applications. Nodes wake up for about 15 seconds for an experiment run of 1800 seconds. As anticipated, nodes 9 and 13 have little higher radio wake-up times due to the transmission of more preambles.

9.2.2 End-to-End Delay

In Figure 9.6, we present an analysis of the average end-to-end data packet delay for AREA-MAC nodes. Unlike the delay analysis presented in Chapter 7, where packet delay at each node has been drawn, we present the average end-to-end delay

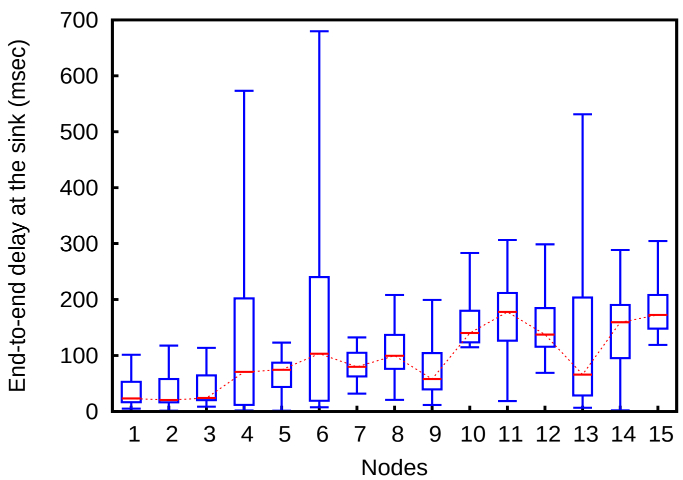


Figure 9.6: End-to-end packet delay at the sink node with respect to data packets generated by other DES WSN nodes for the AREA-MAC implementation on the DES-Testbed platform.

at the sink node with respect to every individual node in this figure. In other words, we assess the time data packets that are generated by each node take to arrive at the sink node. The more data packets a node transmits (and/or forwards), the higher is its average end-to-end delay at the sink node. Nodes 6, 8, 10 - 12, and 14 - 15 not only transmit data packets generated by themselves but also serve as forwarding nodes for other nodes, hence they have marginally higher packet delays as compared to other nodes. On the whole, AREA-MAC provides extremely low average end-to-end packet delay ($<$ 100 ms) for the DES-Network.

9.2.3 Packet Delivery Ratio

Figure 9.7 presents an analysis of data packet transmission and reception of DES-nodes working under AREA-MAC protocol. Data packet generation and overall packet delivery ratio at the sink node have also been included in this analysis. Since the sink node does not generate and send data packets, we do not include it in the data generating and transmitting evaluation.

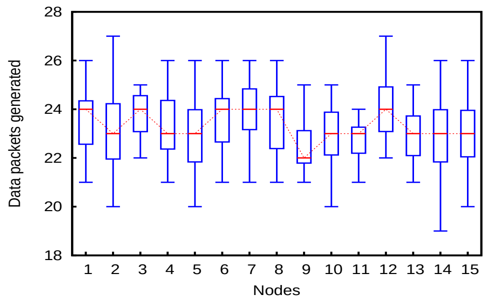
Figures 9.7a demonstrates the number of data packets generated by each node. Since nodes generate data packets with the same generating and deviation intervals as with the simulation evaluation in Chapter 7, they generate between 20 to 29 data packets for an experimentation run of 1800 seconds, the same way as they do with simulation.

The number of data packets received by nodes is outlined in Figure 9.7b. In accordance with the required design goal of the protocol, the sink node receives a significantly higher number of data packets, whereas other nodes receive quite a less number of data packets. It is unlike the traditional long preamble scheme, where almost all nodes receive a large number of data packets due to the transmission of very long preamble prior to each data packet that enables every neighbor node to receive the subsequent data packet (Figure 7.7i). Hence, nodes with the long preamble scheme consume more energy in receiving, processing, and then forwarding duplicate data packets.

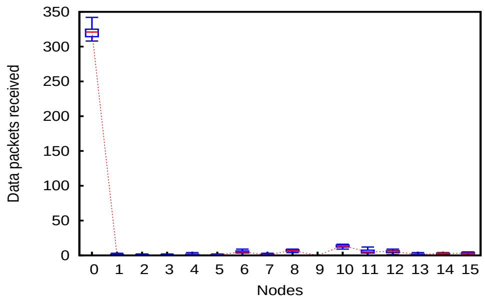
Nodes with IDs 6, 8, and 10 - 11 receive comparatively more data packets than other AREA-MAC nodes because they serve as forwarding nodes for their neighbors, particularly node 10 that receives most of the data packets transmitted by node 9.

In Figure 9.7c, we calculate the number of data packets that successfully arrive at the sink node from each individual node. The sink receives more than 90% of packets generated by each node except the node 9. Apart from the hardware issue, node 9 is also multi-hop (most of the time more than two-hop) away from the sink node and therefore completely relies on its neighbors to forward its data packets to the sink node. Few of its data packets are being dropped during multi-hop communication before arriving at the sink node.

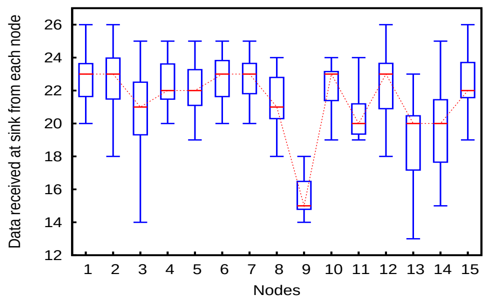
The number of data packets transmitted by each node, which is the sum of data packets generated by the node itself and forwarded by it, is depicted in Figure 9.7d. Each node, on average, transmits between 25 and 30 data packets except node 9



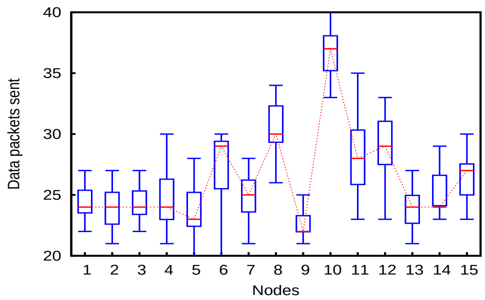
(a) MSB-A2: Data Packets Generated



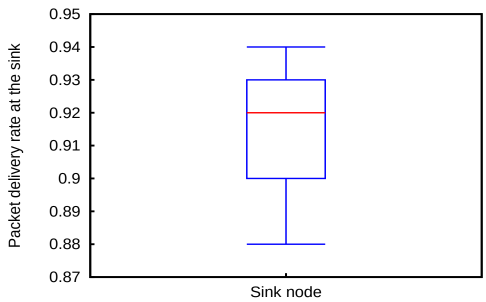
(b) MSB-A2: Data Packets Received



(c) MSB-A2: Data Packets Received at sink



(d) MSB-A2: Data Packets Sent



(e) MSB-A2: Packet Delivery Ratio

Figure 9.7: An analysis of the packet delivery ratio including data packet generation, transmission, and reception of MSB-A2 nodes working under AREA-MAC protocol. (a) depicts the data packet generated by each node, (b) depicts the data packet received at each node, (c) depicts the data packets received by the sink node with respect to each individual node, (d) depicts the data packet sent by each node, and (e) depicts the total packet delivery ratio at the sink node for the DES-Testbed implementation of AREA-MAC protocol.

and its surrounding neighboring nodes like nodes with IDs 8 and 10 - 12. Node 9 transmits a less number of data packets as it does not receive any single packet from its neighbors, whereas node 10 sends out a maximum number of packets (almost 50% more than other nodes) including the most generated packets of node 9.

The last data evaluating Figure 9.7 demonstrates the overall packet delivery ratio at the sink node for the AREA-MAC network. The sink node receives more than 90% of the data packets that are being generated by other nodes for the DES-Testbed implementation. This packet delivery ratio is little less (mainly due to the node 9) than the one achieved with the simulation implementation of AREA-MAC protocol (Figure 8.3). However, it is quite an acceptable amount for a distributed, random, and multi-hop deployment of nodes in a real scenario with much less energy consumption and packet delay as compared to the long preamble scheme.

9.2.4 Preambles Transmitted

Since preambles are one type of control packets and consume radio energy, one of the main targets of AREA-MAC protocol is to reduce the number of preambles that sensor nodes transmit or receive. The higher amount of energy consumption with the long preamble scheme is mainly due to the employment of extended preambles (Section 7.2.1.4).

In Figure 9.8, we present an analysis of the number of preambles that DES-nodes transmit prior to data packet transmission. This figure confirms the reduced number of transmitted preambles for AREA-MAC nodes as compared to the long preamble scheme (Figure 7.8b). Node 9, which has some location as well as hardware issues, however, needs to send out a significantly higher number of preambles until the respective neighbors wake up, respond to it, and receive its data pack-

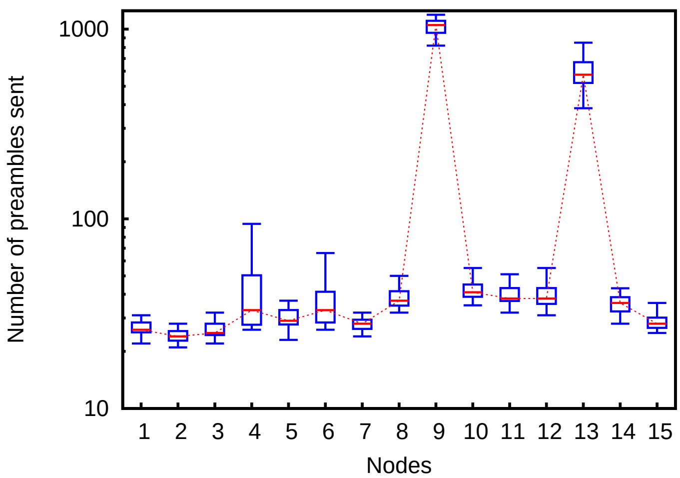


Figure 9.8: Number of preambles transmitted by each DES WSN node for the AREA-MAC implementation on the DES-Testbed platform.

ets. Similarly, node 13 that has only one neighbor (node 14) transmits many more preambles until node 14 wakes up and accepts its data packets. All the other nodes transmit mostly less than 50 preambles to transmit and forward all of their generated and received data packets. Nodes that are at the multi-hop distance from the sink node understandably transmit a little higher number of preambles than the nodes that are single-hop away from the sink node.

9.2.5 Preamble Received

Figure 9.9 reasserts that the number of preambles received by DES-nodes working under the AREA-MAC protocol is several time less than the nodes working under the long preamble scheme that is shown in Figure 7.8e. Since normal AREA-MAC nodes receive much less number of data packets than the sink node, they also receive very few preambles. Nodes that are closer to the sink node receive a negligible amount of preambles (mostly less than 5 preambles) as sink node being in constant receive mode captures all the surrounding preambles. However, nodes that are at the multi-hop distance from the sink node receive relatively a higher number of preambles, especially neighboring nodes of node 9. Node 9 itself hardly receives any preamble that confirms some hardware issues with it. However, node 9 sends out a large number of preambles that are either received by node 8 or node 10 (or by both). Therefore, both nodes 8 and 10 receive a maximum number of preambles that is almost equal to 20.

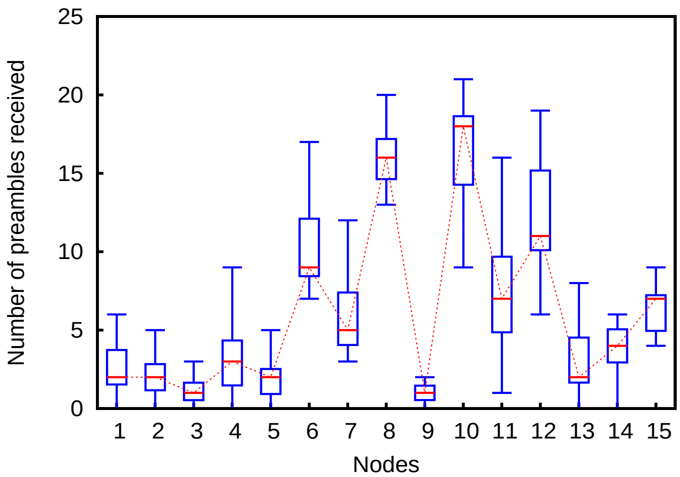


Figure 9.9: Number of preambles received by each DES WSN node for the AREA-MAC implementation on the DES-Testbed platform.

9.2.6 Overheard Packets

For better energy efficiency, sensor nodes must reduce the number of overheard or irrelevant packets that they receive and process. Figure 9.10 validates the reduced number of overheard packets that AREA-MAC nodes receive for the DES-Testbed implementation. Note that an overheard packet for DES-nodes can be a preamble, data, or pre-ACK frame depending on what the receiving node was waiting for and what it has received. Duplicate packets (packets that have already been treated by this node) and loop back packets (packets that have been generated by the node itself) are also counted as overheard packets.

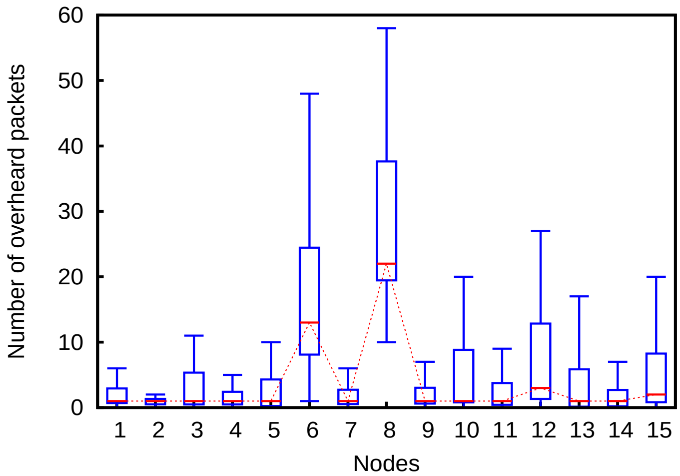


Figure 9.10: Number of overheard packets received by MSB-A2 nodes for the AREA-MAC implementation on the DES-Testbed platform.

As expected, nodes that are at the single-hop distance from the sink node receive almost a negligible number of overheard packets (on average less than 3), whereas farther nodes from the sink node receive a marginally higher number of overheard packets. Node 8 receives a maximum number of overheard packets (up to 40) due to the presence of node 9 in its neighborhood. Node 9, which relies on its neighbors to send out its packets, transmits a large number of preambles in Figure 9.8, and node 8 receives most of them. Similarly node 6 that is surrounded by four other nodes 3, 4, 5, and 7, receives relatively a higher number of overheard packets.

On a whole, AREA-MAC protocol offers much improved results for each considered metric under the real testbed implementation as compared to the long preamble scheme, which also largely validates the simulation implementation of AREA-MAC protocol.

9.3 Simulation Evaluation vs. Testbed Evaluation

In this section, we attempt to concisely equate the performance of AREA-MAC protocol for both simulation and testbed platforms. Note that our main aim is not to provide a comprehensive comparison between the simulation and testbed evaluating platforms but to analyze the performance of AREA-MAC for a wide range of scenarios in order to asses its eligibility for diverse applications. Therefore, we have used varying network conditions for each experiment whether it has been performed on the simulation or testbed platform.

Apart from the analytical analysis (Chapter 6), AREA-MAC protocol has been implemented and evaluated for both the OMNeT++ simulation [48] and DES-Testbed [49] platforms. The rationale behind the selection of OMNeT++ simulator was in fact the first step towards achieving simulation results close to the real implementation. OMNeT++ simulator provides several WSN-specific libraries and models, reduces system complexity, improves system efficiency (in terms of resources and time), offers scalability easiness, and performs better than the well-known *ns simulator* [104, 129]. Moreover, the support of Mobility Framework [103] enables AREA-MAC to use practicable radio, channel, and noise models for a distributed network. Chapters 7 and 8 extensively evaluate the performance of AREA-MAC protocol for several situations including varying topologies, network sizes, routing schemes, and traffic conditions.

In this chapter, the evaluation of AREA-MAC protocol for the real testbed has been presented. Though we have used a smaller portion of the testbed with limited number of nodes and experiment runs due to several reasons, the evaluation of the protocol still gives a very crucial insight into the prefatory comparison of simulation and testbed results. Since the testbed implementation of AREA-MAC is based on the random topology, we mainly compare its results with the simulation implementation of the protocol for the random topology presented in Chapter 7. The simulation configuration of AREA-MAC has been adjusted (in terms of transmission power and channel accessing) in order to efficiently represent the physical obstacles (e.g., walls, rooms without nodes, corridor, etc.) of the real testbed.

By considering and analyzing results obtained from both platforms, we do not discover large divergences for almost all the considered metrics such as energy consumption, packet delay, packet delivery, preamble transmission and reception, and message overhearing. The marginal difference between results for both platforms is in fact inevitable even for an indistinguishable system configuration. AREA-MAC though does not use perfectly identical configurations for both platforms (Table 9.3 points out some of the substantial differences between both implementations), the achieved results still have close matches with each other. For example:

- The testbed implementation of an AREA-MAC network provides less than 0.1% duty cycle for nodes (Figure 9.5d). This value is close to the one that has been achieved with the simulation implementation of AREA-MAC for the random topology (Figure 7.5h). The slight improvement with the testbed implementation is due to the smaller check interval durations of nodes.

- The end-to-end delay at the sink node of the DES-Testbed implementation in Figure 9.6 is very close to that of the simulation implementation of AREA-MAC protocol in Figure 7.6b. The marginal difference of 0.15 seconds in favor of testbed implementation is mainly due to the sink location and lower check interval values of nodes.

- The data packet delivery performance for both platforms is also very close to each other. Since nodes generate data packet at the same interval and deviation rates for both implementations, they generate in total a very close number of data packets for simulation and testbed implementation in Figures 7.7b and 9.7a respectively. However, the number of data packets sent (including the forwarded packets) is almost double for the simulation implementation (Figure 7.7e) than the testbed implementation (Figure 9.7d) because nodes receive more packets from their neighbors due to the fact that many nodes are at the multi-hop distances from the sink node. The total packet delivery ratio for both the simulation (around 96% in Figure 7.7j) and testbed implementation (around 93% in Figure 9.7e) are very close to each other.

- The location of the sink node and the value of check interval affect the number of preambles to be transmitted and received by nodes for AREA-MAC protocol. AREA-MAC nodes transmit and receive about 10 times more preambles in Figures 7.8b and 7.8e for the simulation implementation than the testbed implementation in Figures 9.8 and 9.9 respectively.

- Lastly, the testbed implementation of AREA-MAC protocol results in slightly less overheard packets for nodes in Figure 9.10 than the simulation implementation in Figure 7.9b primarily due to the change in sink location.

Shortly, the central location of the sink node and the selection of suitable check interval value of 500 ms (according to the optimal solution achieved in Chapter 8) ensue a slightly better performance of AREA-MAC protocol for the DES-Testbed over OMNeT++ simulator. On the whole, both platforms have quite a close match

Table 9.3: Major network configuration differences between simulation and testbed implementation of AREA-MAC protocol

Parameter	*Simulation*	*Testbed*
Topology	Random with sink almost at corner	Random with sink almost at center i.e., having maximum node degree
Check interval duration	1000 ms	500 ms
Transmission power	0 dBm	+5 dBm
CCA Threshold	-90 dBm	-91 dBm
WOR Event1	NA	1.3 ms (Radio waits for this time before it turns into the RX mode after waking up)

with each other, which advocates the use of proper system configuration, especially for the simulation environment.

9.4 Conclusion

In this chapter, the real-world implementation and performance evaluation of AREA-MAC protocol on the DES-Testbed platform has been demonstrated. At the beginning, an implementation setup with parameter selection and topology formation has been introduced. Subsequently, several testbed results for energy consumption, packet delay, packet delivery, preamble transmission/reception, and overheard packets metrics have been drawn. The achieved results validate the efficiency of the protocol for all the mentioned metrics and confirm the fulfillment of its design objectives. Like the simulation evaluation, the testbed evaluation of AREA-MAC protocol provides ultra low duty cycle for nodes and extremely low end-to-end packet delays with higher than 90% of successful packet delivery at the sink node. Moreover, the number of preambles transmitted and received and the number of overheard packets for AREA-MAC nodes have also been significantly less for the testbed implementation.

At the end of the chapter, both simulation and testbed evaluation have been matched with each other, which establishes a close equality between them. In fact, the testbed implementation offers slightly improved performance for most of the metrics due to the selection of proper check interval duration for nodes and the proper location of the sink node. The capability of the protocol of performing well between all three evaluation methods counsels that the gap between these methods, particularly between the simulation and real implementation platforms can be minimized if an efficient and a practicable parameter/model selection is used.

After presenting a detailed evaluation of AREA-MAC protocol, in the next chapter we conclude the overall thesis work, highlight the main contributions of the thesis, and figure out further implementation alternatives and improvements on AREA-MAC protocol.

Chapter 10

Conclusion and Future Work

Experience seems to most of us to lead to conclusions, but empiricism has sworn never to draw them.

– George Santayana

THIS chapter concludes the research work that has been presented in this thesis, highlights the main contributions and results, and outlines directions for the future work.

10.1 Conclusion

Due to the continuing proliferation of wireless sensor networks in many diverse applications, it becomes increasingly important to their communication protocols, particularly MAC protocols not to restrain themselves with the energy efficiency objective. Considering the speedy achievements in producing ultra low power microprocessors and transceivers, we conceive that in order to efficiently utilize sensor networks on a large scale for a long term and in a wide range of applications, energy efficiency, even though being the most critical metric, is not a sufficient challenge to address.

In this dissertation, beside energy consumption, we have considered packet delay, packet delivery, message overhearing, control overhead, scalability, and adaptability problems of WSN MAC protocols by first outlining inadequacies of state of the art techniques and then proposing a new MAC protocol. Through an extensive performance evaluation, we have shown that the proposed asynchronous and scalable scheme offers better energy conservation, very low end-to-end packet delay, effective packet delivery ratio, and modest message overhearing for sensor nodes both on simulation and testbed platforms. The major contribution of this work are outlined below.

- We have presented a comprehensive study of several WSN MAC protocols working under various channel accessing schemes and have analyzed them for several metrics, which shows that most of the protocols lean towards the energy efficiency. They either support other metrics partially or trade them off in favor of energy. We have exploited main reasons of energy consumption of these protocols that advocate that even the energy consumption of most of these protocols can be further improved. Importantly, many of them do not consider latency as an important metric in their scope, which motivates the design of our proposed AREA-MAC protocol.

- We have studied IEEE 802.15.4 MAC protocol in detail and have identified several limitations of the protocol, particularly for energy, latency, and bandwidth related WSN applications. We have proposed a simple yet effective scheme to cope with these limitations. The analytical results show an improved performance for an 802.15.4 network in terms of energy consumption, packet latency, and bandwidth under-utilization over the default 802.15.4 standard. Our scheme prioritizes the channel access of devices as per their delay and bandwidth requirements and helps a coordinator in allocating more real-time slots to more needy devices. Moreover, the proposed scheme significantly reduces energy consumption of network devices including the coordinator.

- As a major contribution to this thesis work, we have designed a novel Asynchronous Real-time Energy efficient and Adaptive MAC (AREA-MAC) protocol for WSNs. AREA-MAC is an asynchronous MAC protocol for WSNs that works on the channel sampling mechanism without the need of any type of synchronization among nodes. It achieves significant gains in terms of energy and delay for sensor nodes by substituting long preambles (of the traditional long preamble scheme) with the short ones and by putting extra information with each preamble. The protocol attains reduced energy consumption in several ways that include reduction in idle listening, overhearing, collisions, over-emitting, and control overhead. AREA-MAC results in improved packet delay for sensor nodes by offering several features such as pre-ACK, pre-sleep, adaptive duty cycling, short and long sleep modes, message prioritizing, and convergecast communication. Moreover, AREA-MAC reduces packet redundancy, increases packet delivery ratio, supports suitable adaptability and scalability, and offers a cross-layer functionality by using different routing and traffic generating schemes for WSNs. All these AREA-MAC features have been outlined in Figure 1.3.

- We have evaluated AREA-MAC protocol with the analytical, simulation, and real experiment methods, and the obtained results clearly satisfy the design goals of the protocol. Results obtained from these evaluating methods, particularly with simulation and testbed platforms match quite closely with each other, which indicates the effectiveness of the used simulation models and configuration. Experimentation results show that AREA-MAC offers ultra low duty cycle values for nodes (around 1% for many considered scenarios), very low end-to-end packet delay ($<$ 100 ms for most of the scenarios), an acceptable packet delivery rate ($>$ 90% for most of the scenarios), significantly less control overhead, and very reduced message overhearing as compared to the traditional long preamble based channel sampling method.

- We have studied the effect of varying duty cycle values on the energy consumption, end-to-end packet delay, and packet delivery ratio of AREA-MAC nodes. Through results, we have analyzed that nodes cannot indefinitely reduce or extend their duty cycle values. To find an optimal duty cycle value of nodes, we have employed the mathematical linear optimization method to formalize three different optimization problems, each to minimize the energy consumption, minimize the end-to-end packet delay, and maximize the overall system lifetime of a WSN. We conclude that AREA-MAC nodes achieve optimal (minimum) energy consumption with the least wake-up time of 23.77 seconds (for an experiment run of 1800 seconds) at the check interval value of 2 seconds. Whereas, the optimal (minimum) end-to-end packet delay of 5 ms has been observed at the check interval value of 0.5 seconds, which also offers better packet delivery ratio (about 98%) for an AREA-MAC network.

10.2 Future Directions

The proposed AREA-MAC protocol has been evaluated with analytical, simulation, and real experimentation methods. Given that it has proven itself for a variety of experimental scenarios, it can be expected to offer strong assistance in investigating the following factors.

- The simulation has provided most extensive evaluation of the protocol along with the comparison with one of the state of the art technique. However, for the greater intensity, it would be very interesting to evaluate the performance of AREA-MAC protocol compared with other channel accessing schemes and/or protocols working under those schemes, and for a more denser/larger network.

- To evaluate the performance of AREA-MAC protocol under real-world conditions, we have implemented and assessed the protocol under the DES-Testbed platform. However, we have used a smaller portion of the testbed with limited number of nodes due to several reasons. It would be exciting to use a larger testbed with some real applications compared with some other schemes in order to gain more profound understanding of the proposed protocol.

- Though we have tried to keep the configuration of both simulation and testbed implementation very close to each other, there are still some difference between them. In order to analyze an exact gap between both platforms, AREA-MAC needs to adjust all simulation models and parameters including the modeling of obstacles as close to real-world as possible.

- We have measured energy consumption of nodes by calculating time its radio remains in receive, transmit, and sleep mode. However, techniques such as energy profiling can be used to know the exact amount of energy consumed by nodes (or left with them) at a given point of time.
- For the cross-layer functionality of AREA-MAC protocol, we have considered a static routing scheme with stationary nodes. It would be realistic to consider a dynamic routing scheme possibly with mobile nodes, so that it can be efficiently analyzed how AREA-MAC nodes dynamically adapt themselves to mobile scenarios.
- The real-world implementation of the protocol has considered MSB-A2 sensor nodes. It would be worthwhile to asses the protocol for other platforms in a heterogeneous environment.

Finally, the proposed scheme to enhance the performance of IEEE 802.15.4 MAC protocol for energy, delay, and bandwidth critical WSN applications needs to be implemented with simulation and/or real-world scenarios, so that the relative analytical results can be asserted.

10.3 Concluding Remarks

The proposed schemes of this thesis take a step in a new direction to support efficient deployment of WSNs in applications such as medical urgency, surveillance, terrorist attacks, home automation, flood, fire, and seismic detection, where energy as well as time are the critical factors. The proposed AREA-MAC protocol attains considerably reduced energy consumption and packet latency and an increased packet delivery for a WSN. Several other positive aspects of the protocol have been analyzed throughout this dissertation. The contents of this thesis explain and document the research endeavors, which we hope will aid research community in future design and implementation of energy and latency bound protocols for wireless sensor networks in a wide range of applications.

References

[1] Ian F. Akyildiz and Mehmet Can Vuran. *Wireless Sensor Networks.* John Wiley and Sons Ltd, 2010.

[2] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of Wireless Sensor Networks, Theory and Practice.* John Wiley and Sons Ltd, 2010.

[3] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks.* John Wiley and Sons Ltd, 2006.

[4] Xiang Yang Li. *Wireless Ad Hoc and Sensor Networks, Theory and Applications.* Cambridge University Press, 2008.

[5] I. Mahgoub and M. Ilyas. *Sensor Network Protocols.* Taylor and Francis Group, 2006.

[6] Sudip Misra, Isaac Woungang, and Subhas Chandra Misra. *Guide to Wireless Sensor Networks.* Springer-Verlag London Limited, 2009.

[7] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless Sensor Networks, Technology, Protocols, and Applications.* Wiley Interscience Publications, 2007.

[8] M. Tubaishat and S. Madria. Sensor Networks: an overview. *IEEE Potentials, 22(2):20–23*, April 2003.

[9] Tmote Sky data sheets. `http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf`.

[10] MSB-A2 data sheet. `ftp://ftp.inf.fu-berlin.de/pub/reports/tr-b-08-15.pdf`.

[11] Imote2 data sheets. `http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=134%3Aimote2`.

[12] John A. Stankovic, Tarek Abdelzaher, Chenyang Lu, Lui Sha, and Jennifer Hou. Real-time communication and coordination in embedded sensor networks. In *IEEE*, volume 91, pages 1002–1022, July 2003.

[13] Pardeep Kumar and Mesut Güneş. Medium Access Control Protocols for Wireless Sensor Networks: Design Space, Challenges, and Future Directions. In Dr. Khalid Ragab, Dr. Azween B Abdullah, and Noor Zaman, editors, *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management, to be appear*. IGI Global Publisher, 2011.

[14] Kurtis B. Kredo II and Prasant Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, 51(4):961–994, 2007.

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

[16] K.G. Langendoen. Medium access control in wireless sensor networks. In H. Wu and Y. Pan, editors, *Medium Access Control in Wireless Networks*, pages 535–560. Nova Science Publishers, Inc., May 2008. ISBN 1-60021-944-6. URL `http://www.st.ewi.tudelft.nl/~koen/papers/mac4wsn.pdf`.

[17] Ilker Demirkol, Cem Ersoy, and Fatih Alagoez. MAC Protocols for Wireless Sensor Networks: A Survey. *IEEE Communications Magazine*, April 2006.

[18] SunSPOT data sheet. `http://www.sunspotworld.com./docs/Yellow/eSPOT8ds.pdf`.

[19] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43:51–58, May 2000.

[20] MicaZ data sheets. . `http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=148%3Amicaz`.

[21] Mica2 data sheets. . `http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=147%3Amica2`.

[22] IRIS data sheets. `http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135%3Airis`.

[23] TelosB data sheets. `http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=152%3Atelosb`.

[24] Memsic. `http://www.memsic.com`.

[25] Anna Hac. *Wireless Sensor Network Designs*. John Wiley and Sons, Ltd., 2003.

[26] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.

[27] W. Ye, J. Heidemann, and D. Estrin. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Trans. Net.*, 12(3):493–506, June 2004.

[28] T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 171–180, November 2003.

[29] P. Lin, C. Qiao, and X. Wang. Medium Access Control with a Dynamic Duty Cycle for Sensor Networks. In *IEEE WCNC*, volume 3, pages 1534–1539, March 2004.

[30] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*, 1(4), October 2002.

[31] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. In *ACM SenSys, Los Angeles, CA*, November 2003.

[32] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J Pottie. Protocols for Self-Organization of a Wireless Sensor Network. In *the 37th Allerton Conference on Communication, Computing and Control*, September 1999.

[33] Chris Baker, Kenneth Armijo, Simon Belka, and et al. et al. Wireless Sensor Networks for Home Health Care. In *21st IEEE International Conference on Advanced Information Networking and Applications Workshops*, Washington, DC, USA, 2007.

[34] Aleksandar Milenkovic, Chris Otto, and Emil Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications (Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond*, 29:2521–2533, 2006.

[35] Nada Golmie, David Cypher, , and Olivier Rebala. Performance Evaluation Of Low Rate WPANs For Medical Applications. *Military Communications Conference*, October-November 2004.

[36] The Internet of Things. `http://www.itu.int/osg/spu/publications/internetofthings/`.

[37] The HP's CeNSE Project. `http://www.hpl.hp.com/research/intelligent_infrastructure/`.

[38] The IBM's Smarter Planet Campaign. . http://www.ibm.com/smarterplanet/us/en/.

[39] Kevin Klues, Gregory Hackmann, Octav Chipara, and Chenyang Lu. A Component-Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks. In *The 5th International Conference on Embedded Networked Sensor Systems (SenSys'07)*, November 2007.

[40] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* John Wiley and Sons Inc., New York, US, April 1991.

[41] Klaus Wehrle, Mesut Günes, and James Gross. *Modeling and Tools for Network Simulation.* Springer Verlag, 2010.

[42] IEEE 802.15.4-2006: MAC and PHY Specifications for LR-WPANs. http://ieee802.org/15/pub/TG4.html.

[43] Moteiv Tmote. http://www.sentilla.com/pdf/eol/tmote-sky-brochure.pdf.

[44] TinyOS 2.x. http://www.tinyos.net/tinyos-2.x.

[45] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD*, pages 95–107, November 2004.

[46] Amre El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks. *IEEE International Conference on Communications (ICC)*, 5:3418–3423, 2002.

[47] A. El-Hoiydi and J. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. In *Ninth IEEE Symposium on Computers and Communication, ISCC04*, pages 244–251, June 2004.

[48] OMNeT++: Objective Modular Network Testbed in C++. . http://www.omnetpp.org.

[49] DES-Testbed. http://www.des-testbed.net.

[50] Intel Xscale Mircoarchitecture. http://www.intel.com/design/intelxscale/xscaleproductbriefweb.pdf.

[51] NXP Microcontroller. http://ics.nxp.com/literature/other/microcontrollers/pdf/newsletter.microcontrollers.pdf.

[52] MSP Microcontroller. http://focus.ti.com/lit/ds/symlink/msp430f1101a.pdf.

[53] ATMEL AVR Microcontrollers. http://www.atmel.com/products/avr/default.asp?family_id=607&source=sec_nav.

[54] Chipcon Radios. http://focus.ti.com/analog/docs/enggresdetail.tsp?familyId=367&genContentId=3573.

[55] MPR2400 Radio. http://www.willow.co.uk/html/mpr2400-_micaz_zigbee.html.

[56] RFM TR Radio. http://www.rfm.com/products/data/tr3000.pdf.

[57] Jun Zheng and Abbas Jamalipour. *Wireless Sensor Networks, A Networking Perspective.* John Wiley and Sons, Inc., Hoboken, New Jersey, 2009.

[58] Gilman Tolle, Joseph Polastre, Robert Szewczyk, and et al. et al. A macroscope in the redwoods. In *3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA, November 2005.

[59] Vladimir Dyo, Stephen A. Ellwood, and et al. et al. Evolution and Sustainability of a Wildlife Monitoring Sensor Network. In *SenSys'10*, Zurich, Switzerland, November 3-5 2010.

[60] D. Goense, J. Thelen, and K. Langendoen. Wireless sensor networks for precise Phytophthora decision support. In *5th European Conference on Precision Agriculture (5ECPA)*, Uppsala, Sweden, June 2005.

[61] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *First ACM International Workshop on Wireless Sensor Networks and Application (WSNA)*, Atlanta, GA, USA, September 2002.

[62] A. Arora, P. Dutta, S. Bapat, and et al. et al. A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46:605–634, 2004.

[63] G. Simon, M. Maroti, A. Ledeczi, and et al. et al. Sensor network-based countersniper system. In *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, USA, November 2004.

[64] The Dust Network Project. . http://www.dustnetworks.com.

[65] The VigilNet Project. http://www.cs.virginia.edu/wsn/vigilnet.

[66] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *ACM SIGOPS Operating Systems Review*, 2002.

[67] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. In *IEEE Internet Computing*, March/April 2006.

[68] E. A. Basha, S. Ravela, and D. Rus. Model-based monitoring for early warning flood detection. In *ACM SenSys'08*, Raleigh, NC, USA, November 2008.

[69] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. CodeBlue: an Ad Hoc sensor network infrastructure for emergency medical care. In *Workshop on Applications of Mobile Embedded Systems*, Boston, MA, USA, June 2004.

[70] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava. NAWMS: Nonintrusive Autonomous Water Monitoring System. In *ACM SenSys'08*, Raleigh, NC, USA, November 2008.

[71] J. Paek, K. Chintalapudi, J. Cafferey, R. Govindan, and S. Masri. A wireless sensor network for structural health monitoring: performance and experience. In *2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, Sydney, Australia, May 2005.

[72] M. A. Labrador and P. M. Wightman. *Topology Control in Wireless Sensor Networks*. Springer Science and Business Media B.V., 2009.

[73] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[74] Vijay Garg. *Wireless Communications and Networking*. Elsevier - Morgan Kaufmann Publishers, 2007.

[75] J. Hill, R. Szewcyk, A. Woo, D. Culler, S. Hollar, and K. Pister. System architecture directions for networked sensors. In *9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLoS IX)*, Cambridge, MA, November 2000.

[76] Vaduvur Bharghavan. MACAW: A Media Access Protocol for Wireless LANs. In *ACM SIGCOMM'94*, London, UK, August 1994.

[77] Tommaso Melodia, Mehmet C. Vuran, and Dario Pompili. The State of the Art in Cross-layer Design for Wireless Sensor Networks. In *Proceedings of EuroNGI Workshops on Wireless and Mobility. Springer Lecture Notes in Computer Science 3883*, July 2005.

[78] Vikas Kawadia and P. R. Kumar. A Cautionary Perspective On Cross-Layer Design . In *IEEE Wireless Communication Magazine*, February 2005.

[79] Pardeep Kumar, Mesut Günes, Qasim Mushtaq, and Jochen Schiller. Performance Evaluation of AREA-MAC: A Cross-Layer Perspective. In *The Fifth*

International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2010), Seattle, USA, April 2010.

[80] Vijay Bhuse and Ajay Gupta. Anomaly Intrusion Detection in Wireless Sensor Networks. *Western Michigan University, Kalamazoo, MI-49008 Report*, January 2005.

[81] Yanjun Li, Chung Shue Chen, Ye-Qiong Song, and Zhi Wang. Real-time QoS support in wireless sensor networks: a survey. In *7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, Toulouse, France, November 2007.

[82] Paolo Santi. Topology Control in Wireless Ad Hoc and Sensor Networks. *ACM Computing Surveys*, 37(2):164–194, June 2005.

[83] Jussi Haapola, Zach Shelby, Carlos Pomalaza-Raez, and Petri Mahonen. Cross-Layer Energy Analysis of Multi-hop Wireless Sensor Networks. In *EWSN'05*, pages 33–44, 2005.

[84] Lodewijk Van Hoesel, Tim Nieberg, Jian Wu, and Paul J. M. Havinga. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. In *IEEE wireless communication*, volume 11, pages 78–86, December 2004.

[85] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE transactions on Mobile Computing*, 1(1):70–80, 2002.

[86] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. ZMAC: a Hybrid MAC for Wireless Sensor Networks. In *3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, November 2005.

[87] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, November 2006.

[88] Gahng-Seop Ahn, Emiliano Miluzzo, Andrew T. Campbell, Se Gi Hong, and Francesca Cuomo. Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks. In *The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, Boulder, Colorado, USA, November 2006.

[89] E-Y. Lin, J. Rabaey, and A. Wolisz. Power-Efficient Rendezvous Schemes for Dense Wireless Sensor Networks. In *IEEE ICC*, June 2004.

[90] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In *SenSys'08*, November 2008.

[91] P Karn. MACA: A New Channel Access Method for Packet Radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, September 1990.

[92] Yuan Li, Wei Ye, and John Heidemann. Energy and Latency Control in Low Duty Cycle MAC Protocols. In *IEEE WCNC*, March 2005.

[93] S. Biaz and Y. D. Barowski. GANGS: An energy efficient MAC protocol for sensor networks. In *Annual Southeast Regional Conference*, pages 82–87, 2004.

[94] J. Polastre. *A Unifying Link Abstraction for Wireless Sensor Networks*. PhD thesis, University of California, Berkeley, October 2005.

[95] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless networks. In *4th ACM Conference on Embedded Networked Sensor Systems (SenSys), Boulder, CO*, pages 307–320, Nov. 2006.

[96] Pardeep Kumar, Mesut Günes, Qasim Mushtaq, and Bastian Blywis. A Real-Time and Energy-Efficient MAC Protocol for Wireless Sensor Networks. *International Journal of Ultra Wideband Communications and Systems (IJUWBCS)*, 1(2):128–142, 2009. doi: 10.1504/IJUWBCS.2009.029002.

[97] M. Avvenuti, P. Corsini, P. Masci, , and A. Vecchio. Increasing the efficiency of preamble sampling protocols for wireless sensor networks. In *1st Mobile Computing and Wireless Communications International Conference (MCWC'06)*, pages 117–122, Amman, Jordan, September 2006.

[98] A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. In *IEEE ICC*, Istanbul, Turkey, June 2006.

[99] Raja Jurdak, Pierre Baldi, and Cristina Videira Lopes. Energy-Aware Adaptive Low Power Listening For Sensor Networks. In *2nd International Workshop on Networked Sensing Systems (INSS)*, pages 24–29, San Diego, CA, USA, June 2005.

[100] Zigbee. `http://www.zigbee.org`.

[101] Mark A. Schulze. Linear Programming for Optimization. *Perceptive Scientific Instruments, Inc.*, 1998.

[102] Linear Optimization - Wikipedia. `http://en.wikipedia.org/wiki/Linear_programming`.

[103] Mobility Framework for OMNeT++. `http://mobility-fw.sourceforge.net/`.

[104] OMNeT++ vs. NS-2. . http://ctieware.eng.monash.edu.au/twiki/bin/view/Simulation/OMNeTppComparison.

[105] ns-2/ns-3. http://www.nsnam.org/wiki/index.php/Main_Page.

[106] Witold Drytkiewicz, Steffan Sroka, and et al. et al. A Mobility Framework for OMNET++. *Telecommunication Networks Group, Technical University Berlin*, January 2003.

[107] Andreas Kuntz, Felix Schmidt-Eisenlohr, Oliver Graute, and Martina Zitterbart. Introducing Probabilistic Radio Propagation Models in OMNeT++ Mobility Framework and Cross Validation Check with NS-2. In *1st International Workshop on OMNeT++ (hosted by SIMUTools 2008)*, Marseilles, France, March 2008.

[108] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental Evaluation of Wireless Simulation Assumptions. In *7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM'04)*, Venezia, Italy, October 2004.

[109] Khalid Abdel Hafeez, Lian Zhao, Zaiyi Liao, and Bobby Ngok-Wah Ma. The Optimal Radio Propagation Model in VANET. In *Fourth International Conference on Systems and Networks Communications*, Porto, Portugal, September 2009.

[110] CC1100 data sheet. . http://focus.ti.com/lit/ds/symlink/cc1100.pdf.

[111] Yunxia Chen and Qing Zhao. On the Lifetime of Wireless Sensor Networks. *IEEE Communication Letters*, 9(11):976–978, November 2005.

[112] Isabel Dietrich and Falko Dressler. On the Lifetime of Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(1), February 2009.

[113] Nok Hang Mak and Seah W.K.G. How Long is the Lifetime of a Wireless Sensor Network? In *International Conference on Advanced Information Networking and Applications (AINA'09)*, May 2009.

[114] Joakim Eriksson, Fredrik Österlind, Niclas Finne, Adam Dunkels, Nicolas Tsiftes, and Thiemo Voigt. Accurate Network-Scale Power Profiling for Sensor Network Simulators. In Utz Roedig and Cormac Sreenan, editors, *Wireless Sensor Networks*, volume 5432, pages 312–326. Springer Berlin / Heidelberg, 2009.

[115] IEEE 802.15 Working Group for WPAN. http://www.ieee802.org/15/index.html.

[116] A. Koubaa, M. Alves, and E. Tovar. GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks. In *14th International Workshop on Parallel and Distributed Real-Time Systems*, April 2006.

[117] Jaeyeol Ha, Wook Hyun Kwon, J.J Kim, Y.H Kim, and Y.H Shin. Feasibility Analysis and Implementation of the IEEE 802.15.4 Multi-hop Beacon Enabled Network. In *15th Joint Conference on Communications and Info*, June 2005.

[118] Jianliang Zheng and Myung J. Lee. A Comprehensive Performance Study of IEEE 802.15.4. *Sensor Network Operations, IEEE Press, Wiley Interscience*, pages 218–237, 2006.

[119] Joseph W. Hoffert, Kevin Klues, and Obi Orjih. Configuring the IEEE 802.15.4 MAC Layer for Single-sink Wireless Sensor Network Applications. *Real-Time Systems class project, Washington University*, pages 218–237, December 2005.

[120] K. Shuaib, M. Alnuaimi, M. Boulmalf, I. Jawhar, F. Sallabi, and A. Lakas. Performance Evaluation of IEEE 802.15.4: Experimental and Simulation Results. *Journal of Communications*, 2(4):29–37, June 2007.

[121] Anis Koubaa, Mario Alves, and Eduardo Tovar. i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks. In *18th Euromicro Conference on Real-Time Systems (ECRTS'06)*, Dresden, Germany, July 2006.

[122] Yu-Kai Huang, Ai-Chun Pang, and Hui-Nien Hung. An Adaptive GTS Allocation Scheme for IEEE 802.15.4. *IEEE transactions on parallel and distributed systems*, August 2007.

[123] Li-Chun Ko and Zi-Tsan Chou. A Novel Multi-Beacon Superframe Structure with Greedy GTS Allocation for IEEE 802.15.4 Wireless PANs. *IEEE Communication Society WCNC*, 2007.

[124] N. Salles, N. Krommenacker, and V. Lecuire. Performance Study of IEEE 802.15.4 for Industrial Maintenance Applications. In *IEEE International Conference on Industrial Technology, ICIT'08*, Chengdu, China, March 2008.

[125] Wanghua Wu, Mihai A. T. Sanduleanu, and John R. Long. 17 GHz RF Front-Ends for Low-Power Wireless Sensor Networks. *IEEE Journal of Solid-state Circuits*, 43(9), September 2008.

[126] Lorenzo Rubio, Juan Reig, and Narcis Cardona. Evaluation of Nakagami fading behaviour based on measurements in urban scenarios. *International Journal of Electronics and Communication*, 61:135–138, 2007.

[127] Joohwan Kim, Xiaojun Lin, Ness B. Shroff, and Prasun Sinha. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Trans. Netw.*, 18(2):515–528, 2010.

[128] CC1100 data sheet. . http://focus.ti.com/lit/an/swra126b/swra126b.pdf.

[129] Ugo Maria Colesanti, Carlo Crociani, and Andrea Vitaletti. On the Accuracy of OMNeT++ in the Wireless Sensor Networks Domain: Simulation vs. Testbed. In *Fourth ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'07)*, Chania, Greece, October 2007.