Energy-efficient Wireless Sensor Network MAC Protocol

Michael Ignatius Brownfield

Dissertation submitted to the Faculty of

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Dr. Nathaniel J. Davis IV, Chairman

Dr. Scott F. Midkiff, Co-Chairman

Dr. Y. Thomas Hou

Dr. C. Patrick Koelling

Dr. Timothy Pratt

March 31, 2006

Blacksburg, VA

Keywords: wireless sensor network, energy efficiency, medium access control (MAC)

Energy-efficient Wireless Sensor Network MAC Protocol

Michael I. Brownfield

Dr. Nathaniel J. Davis, IV, Chairman
Dr. Scot F. Midkiff, Co-Chairman
Electrical Engineering

(Abstract)

With the progression of computer networks extending boundaries and joining distant locations, wireless sensor networks (WSNs) emerge as the new frontier in developing opportunities to collect and process data from remote locations. WSNs rely on hardware simplicity to make sensor field deployments both affordable and long-lasting without maintenance support. WSN designers strive to extend network lifetimes while meeting application-specific throughput and latency requirements. Effective power management places sensor nodes into one of the available energy-saving modes based upon the sleep period duration and the current state of the radio.

This research investigates energy-efficient medium access control (MAC) protocols designed to extend both the lifetime and range of wireless sensor networks. These networks are deployed in remote locations with limited processor capabilities, memory capacities, and battery supplies. The purpose of this research is to develop a new medium access control protocol which performs both cluster management and inter-network gateway functions in an energy-efficient manner. This new protocol, Gateway MAC (GMAC), improves on existing sensor MAC protocols by not only creating additional opportunities to place the sensor platforms into lower power-saving modes, but also by establishing a traffic rhythm which extends the sleep duration to minimize power mode transition costs. Additionally, this research develops a radio power management (RPM) algorithm to provide a new mechanism for all WSN MAC protocols to optimize sleep transition decisions based upon the power and response characteristics of the sensor platform's transceiver. Finally, to extend access to sensor data in remote locations, this

research also validates an innovative wireless distribution system which integrates wireless sensor networks, mobile ad hoc networks (MANET), and the Internet.

This research makes two significant contributions to the state-of-the-art wireless sensor networks. First, GMAC's centralized network management function offers significant energy savings and network lifetime extensions over existing wireless sensor network protocols. The second contribution is the introduction of a wireless sensor radio power management algorithm designed to exploit additional power-saving opportunities introduced with the newest generation of faster sensor platform transceivers.

Dedication

This work is dedicated to my wife, Sally, and our six wonderful children, David, Eliza, Claire, Ryan, Emily, and John. Sally and I take on life's challenges together, and we continue to enjoy the journey and God's blessings.

Acknowledgments

I would like to thank the many people who have contributed to this research endeavor. First, I would like to thank my advisor, Dr. Nathaniel J. Davis, IV. His trust, encouragement, support, and guidance provided an enjoyable atmosphere to pursue knowledge and grow intellectually. Dr. Scott Midkiff, my co-advisor, provided insightful feedback on the technical aspects of this research. I would also like to thank my committee members Dr. Tim Pratt, Dr. Tom Hou, and Dr. C. Patrick Koelling for their support. Scot Ransbottom deserves special thanks for his mentorship in the PhD process and for showing me how much of himself a true friend can give. Of course, since my parents always assume responsibility for the good and the bad, I want to thank them for their tireless efforts in raising me, my brother, and my sisters to honor the Lord, respect other people, and pursue excellence. My father, Lieutenant Colonel (Retired) William I. Brownfield, taught me how to reach beyond my capabilities and serve others, and my mother, Patricia D. Brownfield, taught me how to love and forgive.

This work is based upon the combined efforts of the undergraduate students involved in the Virginia Tech Wireless Sensor Network research group: Almohonad Fayez, Theresa Nelson, Kaveh Mehrjoo, and Yatharth Gupta. Their ideas and team effort significantly contributed to the success of our project.

Wayne Donald and Randy Marchany generously supported our Wireless Sensor Networking Group with the VT IT security lab, state-of-the-art research equipment, and an endless amount of networking experience. Randy consistently provided critical, analytical feedback to ensure that the research remained both realistic and secure. I relied on his expertise and friendship on a daily basis.

I want to thank the Lord for His continual blessings in my life. I put my trust in Him, and He continues to make my paths straight.

My wife, Sally, has contributed countless hours in direct support of this work. For this dissertation and all of the conference papers, Sally polished my “engineer speak” into coherent sentences. We truly set out and completed this journey together.

My children, David, Eliza, Claire, Ryan, Emily, and John, make all of life’s challenges and sacrifices worthwhile.

I appreciate the support from OPNET Technologies, Inc. for the use of their OPNET Modeler simulation program and their impressive training sessions during OPNETWORK 2004 and 2005. I would also like to thank the United States Army and the United States Military Academy for their funding and support of this academic opportunity.

Table of Contents

List of Figures

List of Tables

Glossary of Acronyms

ACK:	Acknowledgement Message
AES:	Advanced Encryption System
AM:	Active (Power Save) Mode
AODV:	Ad Hoc On-demand Distance Vector
AP:	Access Point
ATIM:	Ad Hoc Traffic Indication Message
BSS:	Basic Service Set
BTIM:	Beacon Traffic Indication Map
C4I:	Command, Control, Communications, Computers, and Intelligence
CBC:	Cyclic Block Code
CCA:	Clear Channel Assessment
CDMA:	Code Division Multiple Access
CFP:	Contention-Free Period
CRC:	Cyclic Redundancy Check
CSMA:	Carrier Sense Multiple Access
CSMA-CA:	Carrier Sense Multiple Access with Collision Avoidance
CSMA-CD:	Carrier Sense Multiple Access with Collision Detection
CTR:	Counter Mode
CTS:	Clear To Send
CW:	Contention Window (length)
DCF:	Distributed Coordination Function
DES:	Data Encryption System
DHS:	Department of Homeland Security
DIFS:	DCF Interframe Spacing
DSSS:	Direct Sequence Spread Spectrum
DTIM:	Delivery Traffic Indication Message

DOT:	Department of Transportation
EIFS:	Extended Interframe Spacing
ESS:	Extended Service Set
FCS:	Frame Check Sequence
FDMA:	Frequency Division Multiple Access
FEC:	Forward Error Correction
FFD:	IEEE 802.15.4 Full-Function Device
FRTS:	Future Request To Send
GTIM:	Gateway Traffic Indication Message
GTS:	IEEE 802.15.4 Guaranteed Time Slot
IBSS:	Independent Basic Service Set
IEEE:	Institute of Electrical and Electronics Engineers
IFS:	Interframe Spacing
ISM:	Industrial, Scientific, and Medical
LAN:	Local Area Network
LIFS:	IEEE 802.15.4 Long Interframe Spacing
LLC:	Logical Link Control Module
LPM:	Low-power Mode
LQI:	Link Quality Indication
LR-WPAN:	Low-Rate Wireless Personal Area Network
MAC:	Medium Access Control
MAN:	Metropolitan Area Network
MANET:	Mobile Ad Hoc Network
MCU:	Micro-Controller Unit
MSDU:	MAC-layer Service Data Unit
NAV:	Network Allocation Vector
OFDM:	Orthogonal Frequency Division Multiplexing
OLSR:	Optimized Link State Routing
OSI:	Open Systems Interconnection
PAN:	Personal Area Network
PCF:	Point Coordination Function

PHY:	Physical Layer
PIFS:	PCF Interframe Spacing
PS:	Power save Mode
QOS:	Quality of Service
RF:	Radio Frequency
RFD:	IEEE 802.15.4 Reduced-Function Device
RPM:	Radio Power Management
RSSI:	Receive Signal Strength Indicator
RTS:	Request To Send
RX:	Receive or Receiver
SFD:	Start-of-Frame Delimiter
SIFS:	Short Interframe Spacing
SNR:	Signal to Noise Ratio
TA:	TMAC Adaptive Timeout Period
TBTT:	Target Beacon Transmission Time
TCP:	Transport Control Protocol
TDMA:	Time Division Multiple Access
TIM:	Traffic Indication Map
TSF:	Time Synchronization Function
TX:	Transmit or Transmitter
UDP:	Universal Data Protocol
WDS:	Wireless Distribution System
WEP:	Wired Equivalent Privacy
WLAN:	Wireless Local Area Network
WPAN:	Wireless Personal Area Network
WSN:	Wireless Sensor Network

Chapter 1

Introduction

We make a living by what we get,
We make a life by what we give.
--Sir Winston Churchill

The progressive nature of the Information Age creates increasing demands for processed data, and the consistent fulfillment of Moore's Law produces smaller hardware devices with improved capabilities to gather and process this new data. As world business becomes more mobile and computational applications become widely distributed, wireless networks bridge the gap by making distance and movement seamless. Wireless networks require innovative medium access techniques to share the limited broadcast bandwidth in a fair and efficient manner as computing and communications devices continue to proliferate. Integrating sensors into wireless networks offers the ability for applications to monitor and react to events, but their remoteness also introduces challenges for network control and power management.

This dissertation investigates a new wireless medium access control technique to improve energy efficiency and extend the network lifetime of wireless sensor networks. This chapter provides an overview of the research effort. Section 1.1 states the research problem under investigation. A brief background and motivation for wireless sensor network protocols are presented in Section 1.2. Section 1.3 lists the design goals of the research, and the specific

questions addressed by this research effort are listed in Section 1.4. A brief overview of the methodology used is presented in Section 1.5. Finally, Section 1.6 summarizes the anticipated results and describes the organization of the remaining chapters in this document.

1.1 Problem Statement

The purpose of this research effort is to design, implement, and test a new wireless sensor network (WSN) medium access control (MAC) protocol which coordinates sensor node transmissions to extend sleep durations and conserve energy in memory- and power- constrained devices. The research addresses the specific problem that currently proposed energy-efficient WSN MAC protocols do not provide effective network control mechanisms to maximize sleep durations, minimize idle listening, and limit the amount of cluster control traffic overhead. Contention- and reservation-based protocols address some, but not all, of these sources of energy loss. With minimal control overhead, the Gateway MAC (GMAC) protocol dynamically rotates point coordination duties among all the nodes to distribute the management energy costs, allows the other associated nodes to sleep longer, and extends the lifetime of the network. Additionally, the research investigates a new sensor radio power management algorithm which exploits additional power-saving opportunities introduced with the newest generation of faster sensor platform transceivers. Finally, the research validates a wireless distribution system which integrates WSNs, mobile ad hoc networks (MANET), and the Internet.

1.2 Background and Motivation

Sensor networks monitor phenomena as diverse as moisture, temperature, speed, and location using optical, photo, motion, thermistor and piezoelectric detectors. Since wireless networks operate in a broadcast medium, these networks require a medium access control (MAC) layer to resolve contention in a random multi-access environment. The MAC layer protocols must be sensitive to the specific needs of a wide variety of sensing applications. In an effort to make inexpensive sensors ubiquitous, these sensors tend to have limited processing capability, memory capacity, and battery life. Their network interaction promotes long network lifetime in sensor environments which may be characterized by uniform periodic sampling or long periods of inactivity followed by bursty traffic. In dynamic ad hoc network environments,

wireless sensor networks have the challenge of self-adapting to changes in topology, traffic loads, and existing battery conditions.

The Gateway MAC protocol's innovative architecture is motivated by the requirement for wireless sensor networks to minimize the time radios spend in both the idle and receive modes. Research shows that wireless platform transceivers expend a significant amount of energy receiving on an idle channel [Ste97]. Additionally, the newest generation of short-range sensor platforms integrate transceivers which use more energy in receive than in transmit mode. Future WSN MAC protocol architectures must adapt to this new transmission energy paradigm.

Most existing contention-based WSN MAC protocols attempt to limit transmissions and idle listening, but they fail to prevent the nodes from actively monitoring channel contention periods and reservation protocol packets before transitioning to sleep. Idle listening occurs when a radio monitors an inactive channel. These slotted, contention-based protocols reduce idle radio listening by concentrating the network's data transmissions into smaller active periods, and then transitioning to sleep for the remainder of the cycles. Concentrating the transmissions into a smaller active period increases the probability of collisions, thus wasting precious bandwidth and energy. Existing time division multiple access (TDMA) reservation-based protocols establish fixed time periods for nodes to communicate to eliminate the channel contention and idle listening energy costs. To use the bandwidth effectively, these protocols expend significant energy in exchanging control packets to reallocate unused time slots or require complex algorithms to allocate time slots based upon previous traffic requirements.

GMAC combines the advantages of both the contention- and reservation-based protocols to provide significant energy savings by establishing a centralized node to gather all transmission requirements during a contention-based period, and then coordinating their distributions during a reservation-based, contention-free period. Without any additional overhead, the gateway duties are efficiently rotated to disperse the increased network management energy requirements among all of the nodes.

The emergence of new generation radio platforms with increasing data rates requires the development of sleep transition algorithms to optimize power savings. Previous sensor radio platforms operated at much lower bit rates and powered up more rapidly than the current 2.4 GHz IEEE 802.15.4 standard compliant hardware. In the past, WSN MAC protocols were able to permit radios to completely power down during the transmissions in which a radio was neither

the source nor the destination. One technique called *early rejection* relied on the ability for the radio to partially read a packet header, evaluate the destination address, and transition to sleep [LaH04]. Other protocols rely on reservation-based handshaking in which a sender and receiver both broadcast the duration of an upcoming message to allow other nodes to set virtual carrier sensing timers and transition to sleep [DaL03][YeH02][SiR98]. With increased data rates and limited packet sizes constrained by IEEE 802.15.4 low-rate wireless personal area network (LR-WPAN) standards, sensor platforms may not be able to completely power down, even during maximum-sized packet transmissions. The radio power management (RPM) algorithm investigated in this research incorporates the platform-specific radio transition response times for three low power modes: idle (frequency synthesizer turned off), power down (oscillator turned off), and power off (voltage regulator turned off). This algorithm allows current and future protocols to transition to lower power-saving modes in graduated steps according to the available sleep time indicated in the MAC network allocation vector (NAV) table.

Connecting highway vehicles to the Internet and collecting data from remote sensor networks are emerging fields which provide valuable services to the consumer, commercial, public safety, Homeland Security, and military markets. The symbiotic network explored in this research integrates these networks to provide the following mutual benefits: sensor networks gain data messaging to the Internet, and mobile stations gain Internet access in exchange for their willingness to forward the sensor data. For the traveler, highway Internet access provides web browsing access, email, route directions, roadway conditions, and local area services. In addition to capitalizing on the needs of the traveler, commercial applications also include electronic toll collection, fleet tracking, and on-board vehicular diagnostics reporting.

The Gateway MAC protocol, the radio power management algorithm, and the symbiotic wireless distribution system all work together in extending the lifetime and range of wireless sensor networks.

1.3 Design Purpose

The primary design goal for this research is to develop a MAC layer protocol which increases the network lifetime for wireless sensor networks. The WSN GMAC protocol presented in this research leverages both contention- and reservation-based energy-efficiency techniques to reduce idle listening, message overhearing, and contention energy costs. To

provide immediate service, the GMAC protocol integrates seamlessly into the physical layer described by the IEEE 802.15.4 low-rate wireless personal area network standard.

1.4 Research Questions

This research addresses the following questions:

1. What are the benefits of Gateway MAC?
    a. In terms of energy savings
    b. In terms of extended network lifetime
2. What are latency costs of GMAC?
3. What are the security vulnerabilities of GMAC and how can they be mitigated?
4. What criteria should be applied to optimize radio power management?
    a. Energy costs at each power level
    b. Transition energy costs between each level
    c. Transition latency costs between each power level
5. Can off-the-shelf 802.11b, g, and g+ wireless access points and network adapters provide adequate data exchanges between the roadside and mobile users traveling at highway speeds?
6. Can off-the-shelf 802.15.4 low-rate wireless personal area network-based sensor platforms provide adequate data exchanges between the roadside and mobile users traveling at highway speeds?

1.5 Methodology Overview

The protocol development and comparisons are completed using the Jain *ten-step* performance evaluation methodology [Jai91] which is presented in Chapter 3. The protocol models were developed in OPNET Modeler 10.5 using a top-down approach. The wireless sensor MAC protocol models contain statistic-collecting variables to monitor the energy consumption for each radio state and incorporate transition energy and latency costs. The performance of each protocol is evaluated in terms of network lifetime and energy per bit. Simulation parameters were selected based upon the measurements obtained from state-of-the-art wireless sensor platforms. Factors that were varied in the simulation include the MAC

protocols, the radio power management algorithm, and traffic distributions. Analysis was conducted to verify the testing results. The RPM algorithm transition threshold times were determined by programming each of the radio power state transitions into the Moteiv Tmote Sky and Crossbow MICAz platforms' radios and measuring transition times and energy levels using a calibrated instrumentation circuit and an oscilloscope. Finally, the Symbiotic network interface was validated by installing IEEE 802.11g and IEEE 802.15.4 networks along the Virginia Tech Transportation Institute (VTTI) Smart Road and measuring data exchange rates between the roadside network and a mobile vehicle.

1.6 Organization of Dissertation

First, the following chapter provides an overview of the energy-efficiency challenges and techniques. Next, Chapter 3 further describes the objectives and methodology of the research. Chapter 4 highlights the motivation and system design for GMAC. Chapter 5 details the analytical and OPNET simulation models to compare the wireless sensor network protocols. Chapter 6 examines the analytical and simulation results according to the system performance metrics. Chapters 7 and 8 present the design and performance of the radio power management algorithm and Symbiotic network, respectively. Finally, the research conclusions and recommendations for future research are provided in Chapter 9.

Chapter 2

Overview of Wireless Sensor Network MAC Protocols

Learn from yesterday, live for today, hope for tomorrow.
The important thing is to not stop questioning.
--Albert Einstein

This chapter provides a background review of challenges and related research in energy-efficient wireless sensor network (WSN) protocols. Section 2.1 introduces the capabilities and limitations associated with wireless local area networks (WLAN). Closely related to WLANs, the low-power wireless sensor networks in Section 2.2 provide the motivation for the research. Sections 2.3 and 2.4 establish the architecture within which WSN MAC protocols service the network. Section 2.5 analyzes the existing sources of WSN MAC layer energy losses and presents state-of-the-art techniques to mitigate their effects. To fully consider two important MAC protocol design functions, Sections 2.6 and 2.7 provide background in WSN security and time synchronization. Relevant WSN MAC protocol research is then reviewed in Section 2.8. Section 2.9 discusses sensor network traffic modeling, and Section 2.10 reviews techniques to provide distribution systems for remote sensor networks to recover their data. Finally, Section 2.11 provides a chapter summary.

2.1 Wireless Local Area Networks

Wireless local area networks (WLANs) extend the boundaries of traditional wired local area networks (LANs) by unleashing the constrained flow of wire-line data to saturate the surrounding area. Wireless communication offers significant advantages to both users and network designers. Users gain flexible mobility to work anywhere within radio communication range of a network access point and seamlessly retrieve network resources. Roaming around the network, pervasive devices discover each other and permit users to benefit from context aware applications. Network designers gain tremendous advantages in rapid network building, upgrading, and reconfiguration.

Wireless communications also offer significant network challenges. Since the broadcast medium is shared by many devices and networks, channel controls must be implemented at both the network and station level to facilitate fair, regulated access to the medium. Title 47 of the Federal Communications Commission [FCC01] regulations allocated the three Industrial, Scientific, and Medical (ISM) bands shown in Table 2-1 for unlicensed use under strict power guidelines to prevent interference.

Table 2-1 Industrial, Scientific, and Medical (ISM) bands

Band	Frequency Range
UHF ISM band	902 to 928 MHz
S-band ISM	2.4 GHz to 2.5 GHz
C-band ISM	5.725 to 5.875 GHz

Each band is subdivided into channels with much lower throughput capacity than wired channels. Additionally, the wireless environment introduces significant path loss uncertainty, constantly changing in both the space and time domain. Finally, along with the freedom of open wireless network boundaries, the inability to control both active and passive access to the medium increases network security vulnerabilities.

WLAN networks are implemented using IEEE 802.11 standard compliant devices and are classified as either infrastructure or ad hoc networks. Stations in an infrastructure WLAN network transfer data to a central access point (AP). The AP either forwards the message into a distribution system or relays the message to another station within the AP's basic service set (BSS), all stations associated to a particular AP. Ad hoc WLAN network stations form a peer-to-peer relationship with nodes within communications range to form an independent BSS (IBSS).

Ad hoc networks are appropriate in environments where stations form temporary networks without any existing infrastructure connecting them.

2.2 Wireless Sensor Networks

Wireless sensor networks (WSN) may consist of several to thousands of homogeneous or heterogeneous sensors that share the need to organize for data collaboration or network data collection sink routing. Small system platforms which integrate sensors, processors, and transceivers are referred to as motes. Remote sensing platforms are typically characterized by reduced processing capabilities, limited memory capacities, and fixed battery supplies. The WSN energy consumption falls into three categories: sensing, computing, and communicating. Analysis conducted by Soharabi et al. [SoG00] demonstrates that the communications costs dominate a WSN sensor platform's power budget. WLAN networks were designed to minimize delay and maximize throughput, but they do not provide the energy efficiency demanded by WSN networks. As technology makes the hardware smaller, WSN research continues developing innovative, energy-saving techniques at all network protocol layers in order to engineer sensor platforms which can operate unattended for months or even years. The WSN networks must also be scalable to support extremely dense sensor fields. Applications for energy-efficient WSN networks include homeland defense nuclear/biological/chemical (NBC) sensing, military surveillance, and environmental sensing [MaP02][SzM04][SzP04]. These applications generally work in a self-organizing, clustered environment that supports either a single application or collaborative applications. WSN network design requires tradeoffs in throughput and latency to extend network lifetimes.

2.3 Network Architecture Protocol Stack

This energy-based research focuses on designing an energy-efficient WSN MAC protocol which must properly interface with the other WSN network architectural layers. The network architecture protocol stack comprises five layers which perform a variety of functions to efficiently and effectively transfer data between computers and servers distributed across networks. Dividing network responsibilities into distinct layers with specified interfaces permits network designers to update the software and hardware implementations within each layer in a modular manner. Although adding additional cross-level interface communications may permit

feedback-oriented system optimizations, maintaining the standard interfaces ensures seamless integration during peer-to-peer layer upgrades. The Open Systems Interconnection (OSI) standard reference model lists seven distinct network layers, but a simplified model analyzing only five of the layers shown in Figure 2-1 is commonly applied to network research.

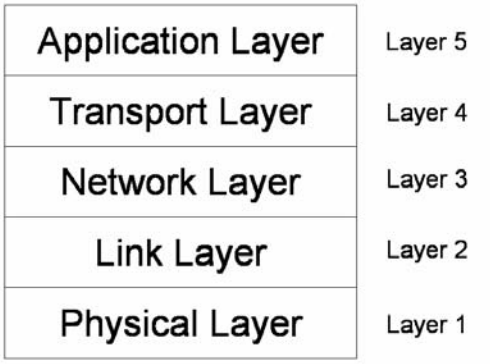

Figure 2-1 Standard Network Architecture

2.3.1 Physical Layer

The physical layer, commonly referred to as Layer 1, interfaces the protocol stack to the network communications hardware to transmit and receive messages one bit or symbol at a time. Individual physical layer functions include Clear Channel Assessment (CCA), data frame synchronization, and encryption. The CCA mechanism senses the physical channel medium to determine if another transmission is currently in progress in order to prevent communications collisions. Data frame synchronization occurs by appending a preamble and start frame delimiter (SFD) sequence of bits to align the radio timing among the network radios. Finally, once the radios are synchronized, the physical layer receives analog symbols from the medium and converts them to digital bits for further processing in higher protocol layers. WSN IEEE 802.15.4 standard radio platforms also offer additional physical layer functions to provide encryption and auto-acknowledgement messages.

2.3.2 Data Link Layer

The data link layer, commonly referred to as the link layer or Layer 2, interfaces with both the network and the physical layer. The link layer is comprised of two separate functions: the Logical Link Control (LLC) module and the Medium Access Control (MAC) layer. The LLC

provides the standard interface between the link and network layer by encapsulating the message segment from the network layer with additional header information that ensures proper sequencing on the distant end. The LLC also assembles and disassembles MAC frames which include data, control information, cyclic redundancy check (CRC) calculation, source address, destination address, and the intermediate network device addresses. For wireless sensor networks seeking to conserve energy, the MAC layer offers significant opportunities to reduce energy and prolong network lifetime since it controls the radio. The MAC layer provides a fair contention mechanism to share access to the medium, data authenticity/privacy security options, and a reliable delivery system using a frame exchange protocol. The frame exchange protocol supports successful delivery by employing acknowledgement frames and data integrity CRC checks. Both the WLAN IEEE 802.11 MAC layer standard and wireless personal area network (WPAN) IEEE 802.15.4 standard are detailed in Section 2.4.2 to elaborate on the MAC layer functionality.

2.3.3 Network Layer

The network layer, commonly referred to as Layer 3, provides the end-to-end routing protocol to connect the sending and receiving stations. Unlike the single-link responsibility of the link layer, the network layer protocol determines the entire delivery sequence of links a packet will traverse across the network on its way to the destination. Based upon the network source or destination address of a packet, the network layer uses algorithmic strategies to optimize the packet routing across various links traversing a network. Wireless, ad hoc networks can proactively send control messages to maintain an accurate routing table or reactively determine a route on demand. WSN networks can use either method, but proactive route discovery techniques like Optimized Link State Routing (OLSR) require additional energy for periodic control messages and memory table allocation for routes that may never be utilized [RFC3626]. The Ad hoc On-demand Distance Vector (AODV) algorithm [IETF04] enables dynamic, self-starting, multi-hop routing between communicating nodes. Since the protocol establishes routes upon request, nodes do not need to maintain routes to inactive nodes. AODV adds additional delay for message delivery, but it ensures a fresh route using minimum power and memory overhead. The AODV algorithm also provides recovery from routes which fail during transmission.

2.3.4 Transport Layer

The transport layer, or Layer 4, offers the ability to regulate traffic flow through the network to the distant end and provide data delivery reliability measures. The transport layer divides large, upper layer application data into sequential segments for delivery. Upon receipt of these segments, the destination's transport layer reorders and reassembles them into data packages for forwarding up to the application layer. Depending on the application requirements, the transport layer can either send out message segments without any reliability mechanisms, or the transport layer can provide flow control, high-level packet error checking, data acknowledgement, and network congestion control. The two standard transport protocol options are transport control protocol (TCP) and universal data protocol (UDP). TCP offers the reliable delivery mechanisms, and UDP maintains simplicity for applications that do not require the control overhead. Unlike the wired transport layer, the wireless transport layer cannot make the assumption that unacknowledged packets are indicative of a congested network. Standard TCP mechanisms which significantly reduce the rate of flow due to perceived congestion will actually cause the system harm in wireless systems.

2.3.5 Application Layer

Network applications provide the purpose for the entire protocol stack. The application layer offers network services directly to the user for electronic mail, file transfers, virtual terminal, and file servers. Network applications use application layer protocols to define the format and order of message exchanges between processes operating on separate networked computers [KuR03].

2.4 Medium Access Control Protocols

As introduced in Section 2.3.2, the medium access control (MAC) layer operates within the data link layer to directly interface with the physical layer to provide fair medium access contention and low-level reliable frame delivery. With its ability to control the radio, this layer affords significant energy-saving opportunities to extend WSN network lifetime. This section

introduces MAC layer challenges, relevant features of MAC standards, classifications of MAC protocols, sources of energy loss in MAC protocols, and techniques to reduce these losses.

2.4.1 Medium Access Control Challenges

The WLAN and WSN MAC layers face significant challenges introduced by the hostile nature of the wireless radio frequency (RF) medium. In addition to the power output restrictions that the FCC places on WLAN radios [FCC01], obstacles, path loss, interference, and multipath signal waves reduce the signal-to-noise ratio and limit the range and data rates for wireless networks. The MAC layer provides the ability to employ error detection and correction codes for data recovery to mitigate these attenuating effects. On-going DARPA research in multi-input/multi-output (MIMO) antennas uses innovative techniques to enable the multipath phase diversity to enhance rather than degrade the signal.

The wireless link presents several access protocol challenges that prevent adopting the same MAC techniques as the wired network. While both wired and wireless LANs use carrier sense multiple access (CSMA) to measure for a clear medium before attempting transmission, only wired LAN transceivers have the ability to detect collisions during transmission by monitoring the voltage level using CSMA with collision detection (CSMA/CD). Unfortunately, WLAN transceivers are unable to simultaneously receive while transmitting. The IEEE 802.11 WLAN MAC standard designates message exchange protocols to indicate successful transmissions by including the return of a positive acknowledgement (ACK) message.

In multi-hop networks, nodes can experience collisions due to an effect called the hidden terminal problem. The dashed lines in Figure 2-2 represent the transmission ranges for nodes 1 and 3. Node 2 is within range of both nodes 1 and 3, but nodes 1 and 3 are not within range of one another. The hidden terminal problem occurs when nodes out of range of one another simultaneously transmit and their messages collide with one another. The collision prevents nodes within both of their radio ranges from successfully receiving either message. For example, if node 1 was sending a message to node 2, node 3 would sense a clear channel since it is out of range of node 1. When node 3 transmits a message, whether or not intended for node 2, the transmission would interfere and corrupt node 1's message to node 2. Hidden node terminal collisions waste valuable energy in retransmissions and increase network congestion. A four-way handshaking collision avoidance (CA) technique is explained in Section 2.4.2.1.4 to

significantly reduce this problem.

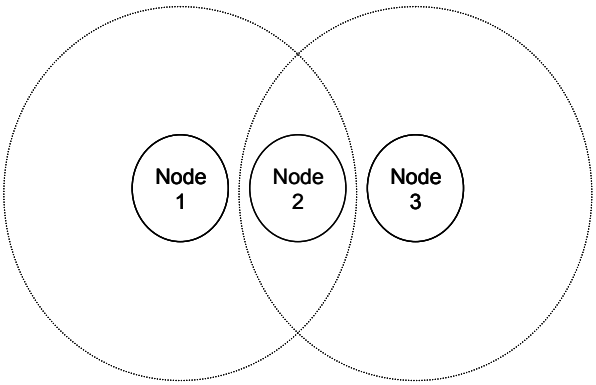

Figure 2-2 Hidden Terminal Problem

2.4.2 Medium Access Control Standards

Standardizing the WLAN and WPAN MAC protocols is an arduous process which involves leading network engineers determining the best methods and time to formalize and distribute an industry standard. This section describes the WLAN and WPAN (Low Rate) techniques which are relevant for energy-efficient protocol design.

2.4.2.1 IEEE 802.11 Wireless Local Area Network Standard

The IEEE 802.11 wireless LAN medium access control (MAC) and physical layer (PHY) specifications were released by the LAN Metropolitan Area Network (MAN) Standards Committee of the IEEE Computer Society in June of 1997 [WLAN97]. Conforming to these standards permits manufacturers to certify their products as “Wi-Fi” [WiFi05].

2.4.2.1.1 IEEE 802.11 WLAN Services

The IEEE 802.11 MAC layer performs nine services: authentication, association, deauthentication, disassociation, distribution, integration, privacy, reassociation, and MAC layer service data unit (MSDU) delivery.

Authentication/deauthentication provides a means to protect the security of the network by verifying the identity of a station attempting to gain access to the network.

Association/disassociation/reassociation are services which register a station to the basic service set (BSS) or group of stations affiliated with a single access point (AP).

Distribution is the service in which an AP accepts an MSDU, evaluates its destination address, and forwards it toward the destination. The distribution system is also responsible for joining access points by bridging them into a backbone network [Gas02].

Integration is the service provided by the distribution system to interface an IEEE 802.11 network with a non-802.11 network in order to transfer MSDUs.

Privacy is a service provided to the WLAN in an effort to obtain equivalent data security as in a wired network. IEEE 802.11 specifies an optional wired equivalent privacy (WEP) to encrypt messages.

2.4.2.1.2 IEEE 802.11 Coordination Functions

Coordination functions control the mechanisms in which stations access the medium. IEEE 802.11 specifies two methods for controlling access: the distributed coordination function (DCF) and the point coordination function (PCF).

2.4.2.1.2.1 Distributed Coordination Function (DCF)

The distributed coordination function (DCF) provides contention-based access to the network. DCF uses a decentralized approach to provide flexibility to the network in responding to variable traffic patterns without any guarantees for quality of service (QOS). Stations use carrier sense multiple access with collision avoidance (CSMA/CA) to sense for a clear channel, send short control frames to reserve the channel and avoid collisions, and then send the actual message. In the event that the channel is not clear due to another transmission in progress, the station will defer access until the channel is clear. Once the channel is clear, all waiting stations will employ a backoff routine designed to reduce collisions in a fair manner.

2.4.2.1.2.2 Point Coordination Function (PCF)

The point coordination function (PCF) augments the DCF functionality and provides contention-free access to the network. Figure 2.3 illustrates that the PCF is an optional function that resides above the DCF in the MAC protocol layer. PCF uses a centralized approach to provide QOS guarantees at the expense of traffic flexibility. Although rarely used, a PCF contention-free period (CFP) begins with the access point (AP) coordinating the period by sending out a beacon frame containing the traffic indication message (TIM). The TIM contains a bit-map listing of all of the stations that the AP has stored traffic to forward and indicates the duration of the CFP. Next, the AP polls each station on the polling list and exchanges messages. All stations on the polling list are required to remain awake during the CFP. A contention-based DCF period follows the CFP to permit communication for stations not participating in polling. Due to the extensive overhead associated with polling, PCF is rarely implemented in WLAN networks.

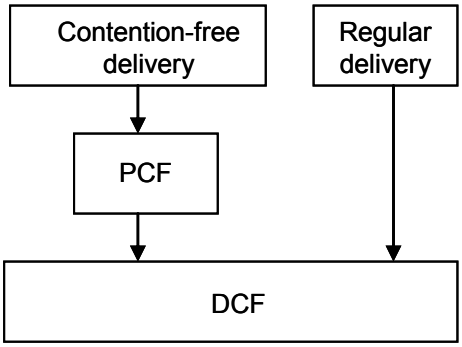

Figure 2-3 DCF/PCF MAC Architecture

2.4.2.1.3 Interframe Spacing

WLAN interframe spacing provides access priority for various classes of messages used in both DCF and PCF operations. For access fairness and congestion control, stations must wait a particular interval after sensing an ideal channel before attempting access. As shown in Figure 2-4, the shorter the deference interval, the higher the priority the station has to gain control of the channel. IEEE 802.11 specifies four distinct time intervals between successive transmissions:

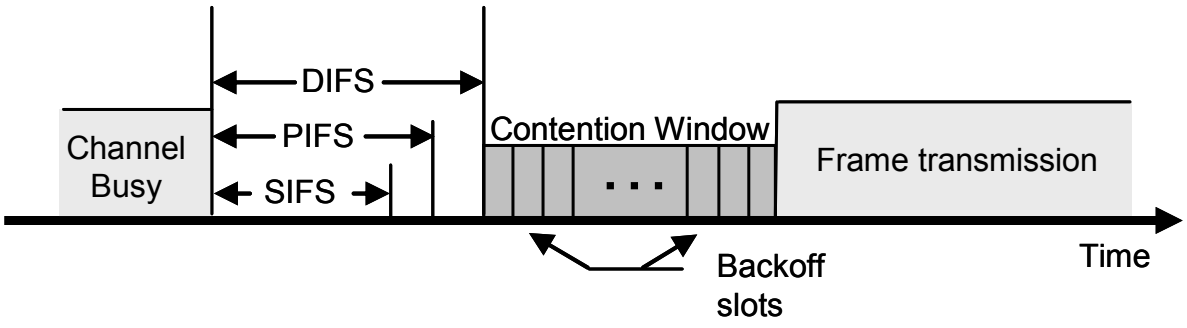

Figure 2-4 Interframe Spacing

SIFS The short interframe spacing (SIFS) represents the highest priority and is typically used to separate successive messages in a single message exchange sequence. Once a station gains control of the medium, SIFS spacing between source and destination stations allows them to maintain control until completion. SIFS are typically used before the acknowledgement (ACK), request to send (RTS), and clear to send (CTS) control messages. Additionally, SIFS are used before follow-on message fragments. IEEE 802.11 defines the SIFS interval spacing as the time between the last symbol of one frame to the first symbol in the preamble of the next frame or the time in which a station can receive and respond to a message:

$$\text{SIFSTime} = \text{RxRFDelay} + \text{RxPLCPDelay} + \text{MACPrcDelay} + \text{RxTxTurnaround} \qquad (2\text{-}1)$$

where:

RxRFDelay represents the delay for processing the signal through the radio receiever,
RxPLCPDelay represents the physical layer convergence protocol (PLCP) delay,
MACPrcDelay represents the MAC processing delay, and
RxTxTurnaround represents the time required to switch the radio from receive to transmit.

PIFS The PCF interframe spacing (PIFS) represents the next to highest priority access deference time. PIFS are used for nodes participating in PCF during non-contention periods. Many research protocols also use PIFS to provide an intermediate priority between SIFS and DIFS.

$$PIFSTime = SIFSTime + SlotTime \tag{2-2}$$

$$SlotTime = CCATime + RxTxTurnaround + AirPropagationTime + MACProcessingTime \tag{2-3}$$

where *CCATime* represents the time for the radio to perform a clear channel assessment (CCA)

DIFS DCF interframe spacing (DIFS) is lower than the PIFS priority and represents the standard contention deference time. Upon the initial attempt to gain medium access, a station must sense the medium as idle for a complete DIFS period before attempting access. If the channel was not idle prior to the DIFS, all stations must execute the contention backoff algorithm to prevent collisions and promote fair access.

$$DIFSTime = SIFSTime + (2 * SlotTime) \tag{2-4}$$

EIFS Extended interframe spacing (EIFS) is the spacing used once a station has detected a frame error reported by the PHY layer when a frame fails the frame check sequence (FCS). This time interval provides the receiving station the opportunity to respond with an ACK if it correctly received the frame.

$$EIFSTime = SIFSTime + (8 \times ACKSize) + PreambleLength + PLCPHeaderLength + DIFSTime \tag{2-5}$$

Figure 2-5 shows a standard DCF atomic message exchange between two stations. The source (Src) initially senses an idle channel and waits the required DIFS interval before taking control of the channel. The source sends a data frame to the destination. The destination (Dest) receives the data and quickly returns an ACK signifying successful receipt. By responding within the SIFS period, the destination prevents other channels from gaining control of the channel. Once all other nodes sense data transmissions, they defer access until the atomic transaction is complete. Since several stations may be waiting to transmit in the next period, they must execute the contention backoff algorithm to prevent collisions.

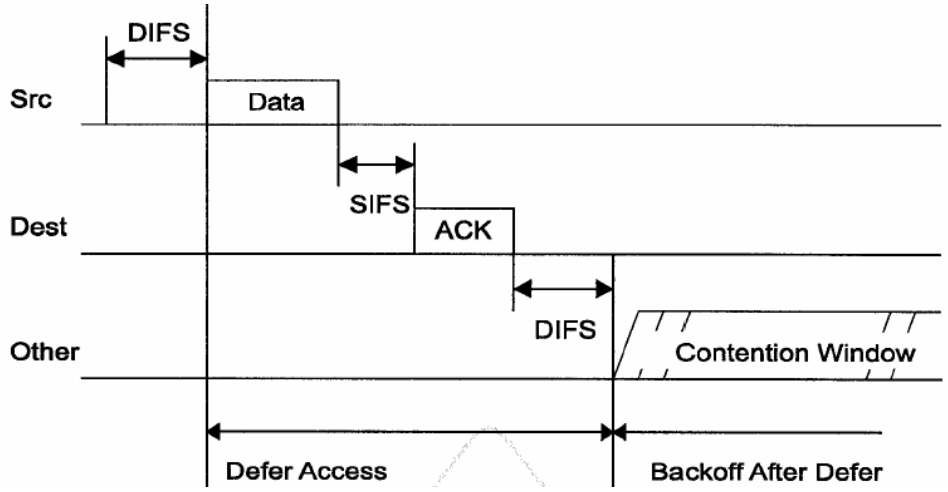

Figure 2-5 Standard Message Exchange [WLAN97]

2.4.2.1.4 CSMA/CA

The carrier sense multiple access (CSMA) protocol utilized in WLAN communications provides a means to prevent collisions by having stations "listen before they speak." Karn [Karn90] developed a request to send (RTS) and clear to send (CTS) handshaking technique to alleviate the CSMA hidden terminal problem described in Section 2.4.1. Bharghavan [BhD94] followed up with a protocol which appends an additional ACK to the RTS/CTS handshake. The IEEE 802.11 committee accepted their ideas as the standard and established CSMA with collision avoidance (CSMA/CA) to produce the RTS-CTS-data-ACK atomic data exchange protocol. Figure 2-6 shows the source node obtaining control of the medium and sending the destination a request to send message (RTS) which reserves the medium for a follow-up transaction. The destination responds with a clear to send (CTS) message. The RTS-CTS provides several advantages to the message exchange. First, the source reserves the channel with a small control frame instead of a larger data frame. If a contention collision occurs, the loss is smaller. Next, the RTS-CTS exchange notifies all nodes within radio range of the source *and* destination. The hidden terminal will receive the destinations CTS frame. Finally, both the RTS and CTS control frames contain a duration field which indicates how long the complete RTS-CTS-data-ACK transaction will take. The duration field allows nodes to set a network allocation vector (NAV) timer to automatically defer access until the announced transaction duration time is complete.

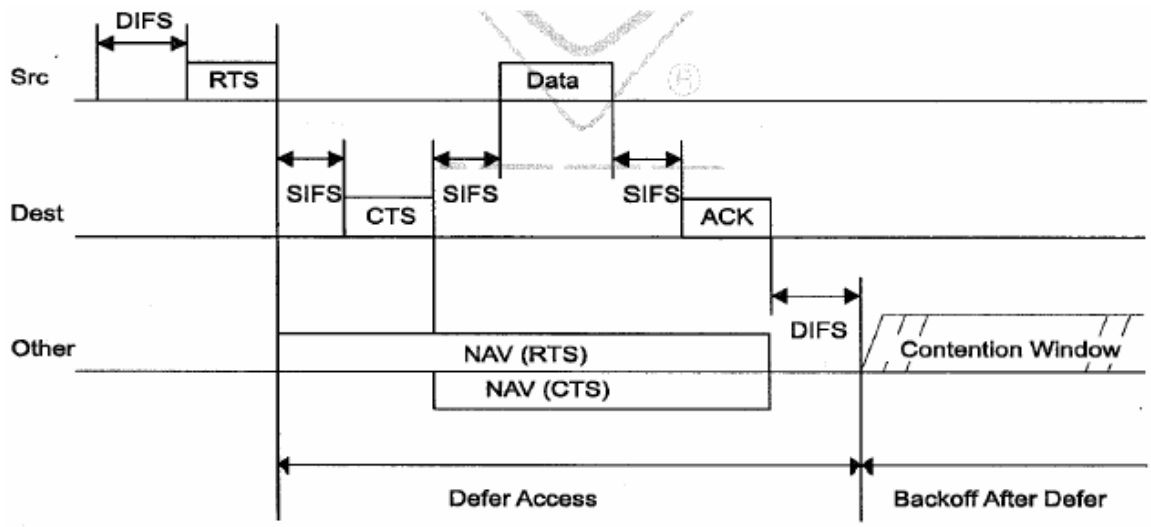

Figure 2-6 RTS-CTS-DATA-ACK and Network Allocation Vector [WLAN97]

2.4.2.1.5 Physical and Virtual Carrier Sensing

CSMA/CA requires that the MAC layer defer contention for the channel while the channel is busy. The PHY layer listens to the channel using a procedure called clear channel assessment (CCA). During CCA, the radio samples the energy level on the channel to determine if the channel is idle. Another means to defer during a busy channel is virtual carrier sensing. In virual carrier sensing, nodes receive the transmission duration field in the RTS-CTS exchange and defer physically sensing the channel until this time has elapsed. The *Other* stations in Figure 2-6 neither receiving or transmitting can employ their NAV timer to indicate the busy channel. In addition to preventing hidden terminals from transmitting before the transaction is complete, virtual carrier sensing also offers a power-saving opportunity for stations to turn off their radios while the other stations are communcating.

2.4.2.1.6 Exponential Contention Backoff

Exponential contention backoff is a DCF mechanism which reduces the probability of frame collisions and stabilizes the network during heavy congestion periods. When a station senses a busy channel, it defers access until the channel is clear for a complete DIFS. Once clear, the station calculates a random number of slot times to wait before attempting to access the channel. If another station calculates a lower random number and seizes the channel before the waiting

station's countdown is complete, the waiting station suspends the countdown counter until the channel has been clear for another DIFS. Once clear, the countdown resumes. At the end of the countdown, the station attempts to access the channel by transmitting its frame. The IEEE 802.11 MAC layer uses the parameters aCWmin and aCWmax to provide the random number boundaries in calculating the exponential backoff. The minimum contention window (aCWmin) is the smallest range of possible backoff slots used in the first backoff. For example, the first backoff interval is Random(0 to aCWmin) * slotTime. If a station experiences a collision during transmission, it increases the size of the contention window by a power of 2, hence the name exponential backoff. Figure 2-7 shows how subsequent collisions continue to increase the range of the random backoff until it reaches the maximum contention window size (aCWmax). If all nodes utilize the same backoff algorithm, the contention process is a fair method to regulate access to the shared medium.

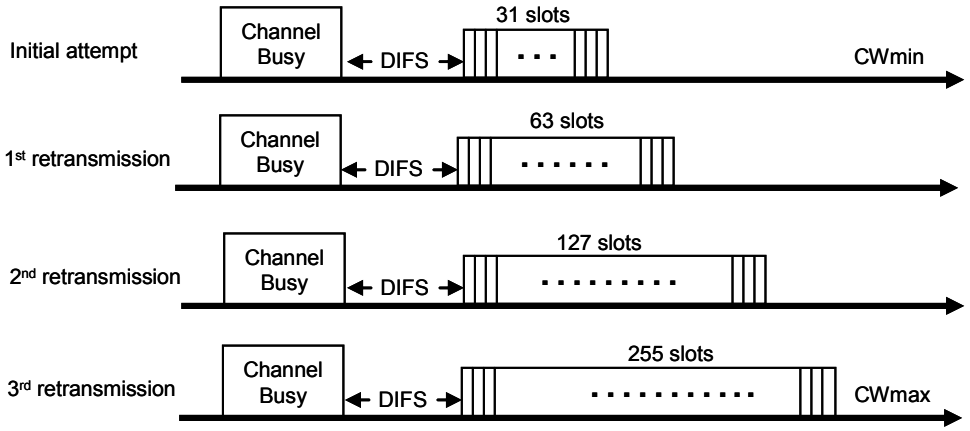

Figure 2-7 Exponential Contention Backoff

2.4.2.1.7 Fragmentation

Fragmentation provides the network manager with a means to control the amount of loss during frame collisions or destructive interference. Based upon message-size thresholds, the MAC layer segments a message into smaller fragments and transmits them one at a time as illustrated in Figure 2-8. The advantage of fragmentation is that when a fragment frame is not successfully received at the destination, the source can retransmit a smaller portion of the original message. Fragmentation may be the only way to successfully transmit data in extremely

noisy environments. The overhead cost of fragmentation is the energy and bandwidth associated in sending/receiving an additional ACK and waiting two additional SIFS times for every fragment sent.

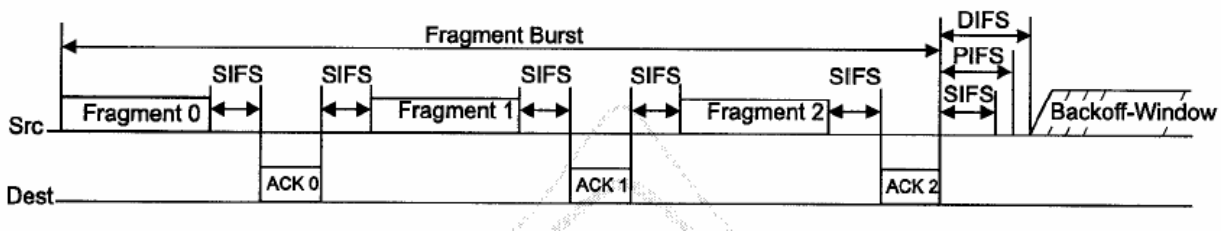

Figure 2-8 Fragmentation [WLAN97]

2.4.2.1.8 DCF, PCF, and Ad hoc Power-saving Modes

Recognizing that many mobile devices have reduced power capabilities, IEEE 802.11 provides power-saving states for stations operating in DCF, PCF, and Ad hoc modes. During the association process, infrastructure-based APs for the DCF and PCF modes use the *Listen Interval* parameter to indicate a max sleep interval for joining stations. Stations can freely trade off packet delivery latency to obtain energy savings by sleeping for this entire duration. If the station exceeds this maximum sleep interval without collecting buffered data, the AP may discard the data.

Stations broadcast a power management bit flag in the frame control field indicating their future active mode (AM) or power save (PS) mode after the current data exchange. This message may be embedded in an actual data packet, a control packet, or a *Null* data packet. A *Null* data packet contains a data MAC header and FCS trailer to indicate a future change in power-saving status [Gas02]. In AM mode, stations remain fully powered and are able to respond to the network. During PS mode, the stations turn off many of their subsystems, including the radio, to conserve power. They are unable to receive or transmit any messages in the PS state. APs maintain a table of all stations within the BSS and their current power-saving state. Since APs vigilantly manage the network, these devices are not permitted to transition into PS mode.

PS mode stations periodically wake up and listen for the AP beacon. Each AP beacon frame is embedded with the AP's traffic indication map (TIM) in order to signal all stations for which

the AP currently has buffered data. The beacon also contains a countdown field with indicates the next Delivery TIM (DTIM). The DTIM is a periodically-spaced frame that immediately sends all buffered broadcast and multicast messages after the beacon. All stations should wake up to receive the unacknowledged broadcast and multicast messages.

PCF stations send the AP a PS-POLL control frame during the DCF period in order to transition from PS to AM modes and become eligible to rejoin the active contention-free polling/distribution list. Once PCF stations signal a transition to AM, the stations remain awake throughout the entire DCF and PCF timeframes.

DCF-only stations in PS mode also periodically wake up and listen for their buffered traffic entry in the beacon TIM field. If they have buffered traffic waiting, the DCF-mode stations send the AP a polling request (PS-POLL) message during the DCF contention period to request delivery. The AP will immediately send the data with the *more data* bit field set to indicate if the AP has additional messages buffered.

Since ad hoc, independent BSS (IBSS) networks do not have a centralized access point, these networks require each station to buffer its own data for sleeping stations. During the DCF contention period, if one station sends an RTS to another station and does not receive a response, the station assumes that the destination is in PS mode. Ad hoc networks self-organize to form an IBSS with their own beacon intervals. At the beginning of each beacon interval, called the target beacon transmission time (TBTT), all stations within an IBSS participate in a contention backoff period to potentially broadcast a beacon. The station with the lowest generated random number in the aCWmin window broadcasts the beacon. Upon receipt of the *winning* beacon, all other stations cancel their scheduled beacon transmissions. The winning node remains awake throughout the next frame. The period immediately following the beacon is the announcement TIM (ATIM) window, also called the ad hoc TIM. The ATIM window provides a contention period opportunity for stations with buffered traffic to send a short ATIM request to a destination node to signal for that station to remain awake after the ATIM window to receive data. All IBSS stations are required to be awake during this ATIM window period. When a destination station receives an ATIM, the station ACKs the ATIM and remains active to receive the message after the ATIM window is complete. Although the duration of the ATIM window is static during the lifetime of the IBSS, Jung and Vaidya present a dynamic ATIM windowing technique to optimize energy savings [JuV02]. The remaining time between the ATIM and the next TBTT is

a DCF contention period to exchange messages. The beacon transmitting station, ATIM message senders, and ATIM message receivers must remain awake throughout the entire DCF contention period.

2.4.2.1.9 Time Synchronization Function (TSF)

Each transmitted IEEE 802.11 beacon frame contains an eight-byte field with the BSS or IBSS timestamp. With each bit capable of representing a microsecond, an eight-byte timestamp would not repeat itself for almost 585 thousand years. Upon successful receipt of the beacon, all stations within the BSS or IBSS will update their time synchronization function (TSF) clocks with an adjusted timestamp value. The timestamp must be adjusted to account for the propagation delay reported by the PHY layer (see [GaK03] for a complete description of determining propagation delay) and the PHY/MAC processing delay. The accuracy of the TSF timer is 0.01% [WLAN97].

2.4.2.2 802.15.4 Wireless Personal Area Network Standard

IEEE 802.15 Subgroup 4 (802.15.4) [WPAN03] establishes wireless MAC and PHY specifications for low-rate wireless personal area networks (WPAN). The purpose of the specification is to "provide a standard for ultra-low complexity, ultra-low cost, ultra-low power consumption, and low data rate" devices [WPAN03]. The data rate can be as high as 250kbs, but scalable to lower data rates for sensor networks.

WPANs operate in a similar manner as the ad hoc WLAN model, where the network must self-configure without a fixed access point (AP). The participating stations within a WPAN are either full-function devices (FFD) or reduced function devices (RFD). Examples of RFDs would be sensor or actuator platforms which operate for a single purpose and have limited protocol functionalities and one-way communications capabilities. The 802.15.4 standard provides MAC layer protocols to self-configure FFDs into a peer-to-peer network. RFDs must link to FFDs to gain network access.

The IEEE 802.15.4 network architecture is based on the OSI model, but simplified down to only the PHY, MAC, network, and application layers. The MAC layer can operate in the DCF mode or elect a FFD PAN coordinator for PCF mode. With a PAN coordinator, the WPAN can divide inter-beacon time frames into contention and contention-free periods using optional

superframes. Shown in Figure 2.9, the superframe contains a contention period with an added quality of service (QOS) feature called guaranteed time slots (GTS). The PAN coordinator can allocate up to seven GTS time slots for low-latency or high-priority application requirements.

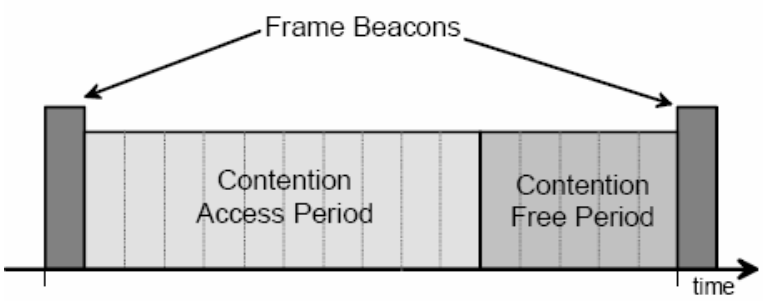

Figure 2-9 802.15.4 Superframe [WPAN03]

The individual PHY protocol data units have a 7-bit frame length field which limits the MSDU maximum size limitation of 128 bytes (1 bit in frame length octet is reserved for future use). Figure 2.10 illustrates the minimalist overhead structure for both the PHY and MAC layer protocol data units. Depending on the size of the addressing field, the data payload size in the MSDU is limited to a maximum of 103 to 119 bytes.

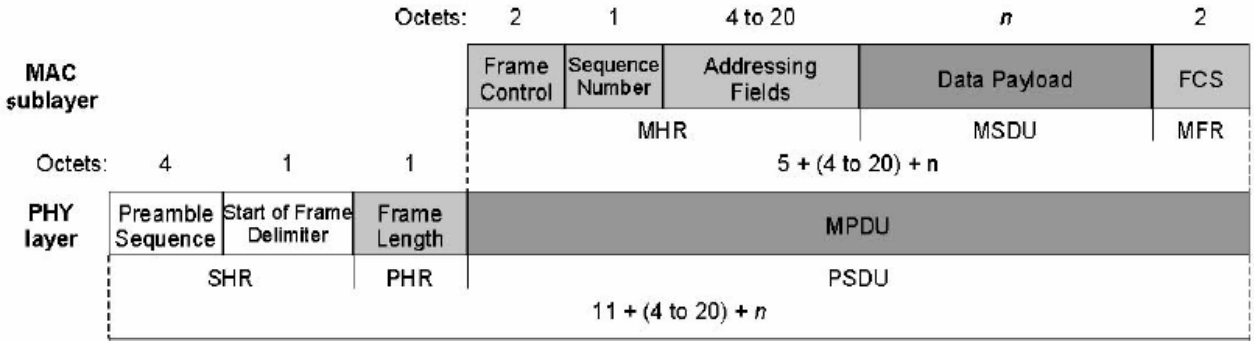

Figure 2-10 802.15.4 Data Frame [WPAN03]

The IEEE 802.15.4 interframe spacing concept is simpler than IEEE 802.11. The WPAN network uses two interframe spacing sizes, short interframe spacing (SIFS) and long interframe spacing (LIFS). The interframe spacing sizes are intended to provide the PHY layer adequate time to process a received message. The IEEE 802.15.4 standard's default SIFS size is the

duration of 11 symbols, and the LIFS size is the duration of 40 symbols. The general rule on which interframe spacing to use depends on the size of the previous frame. If the previous frame was longer than the *aMaxSIFSFrameSize* parameter, the interframe spacing will be a LIFS. Otherwise, the station will only wait a SIFS.

Finally, although IEEE 802.15.4 does not explicitly define a power-saving mode, the standard incorporates some MAC layer parameters for designers to improve on its power efficiency:

macBattLifeExt (Boolean): This parameter is used by the MAC layer to limit the contention period backoff exponent ≤ 2. This range limit gives energy-constrained devices preferential access to the medium.

macBattLifeExtPeriods (Integer): This parameter is used by the MAC layer to limit the number of backoff periods a receiver with the parameter *macBattLifeExt* = TRUE will remain active after the beacon. The number of backoff periods under this condition is either 6 or 8 depending on the physical layer channel settings. This setting allows the network designers to predict the lower boundary of a station's lifetime.

IEEE 802.15.4 WPAN standard uses many of the same functions as IEEE 802.11 WLAN standard and sets very general guidelines for low-rate, low-complexity devices.

2.4.3 Classifications of Medium Access Control Protocols

Wireless sensor network MAC layer protocols fall into a wide variety of classifications depending on how they access the medium, the number of available channels to send data/control messages, their deployment method, their power constraints, quality of service requirements, etc. The more a wireless sensor network resembles existing fixed infrastructure networks, the more contemporary solutions are used to solve the system design challenges. Two broad categories of MAC layer protocols are *Contention-free* and *Contention* MAC protocols. Within each category, methods to control individual access to the shared medium vary according to assumptions about traffic patterns, quality assurance, fairness, and complexity. The following sub-sections describe the advantages and disadvantages of three major protocol categories.

2.4.3.1 Contention-free, Schedule-based MAC Protocols

Contention-free, or schedule-based, protocols operate using node or link activation schemes to provide fair, multiplexed access for every station. These synchronous protocols rely on centralized control functions and work very well when network traffic is evenly distributed. The central coordinator creates schedules either statically or dynamically, with dynamic scheduling requiring much more control message overhead to determine the individual station needs within the network. Contention-free MAC protocols offer the ability to eliminate the contention overhead and reduce the idle listening overhead for energy-constrained networks. By reserving specific time slots or channels for each station, these schedule-based protocols can make quality of service (QOS) guarantees for latency-constrained or high priority applications. Guaranteeing bandwidth also provides stability for networks under heavy loads. During periods of sparse traffic, scheduling protocols tend to waste bandwidth in the form of unused timeslots or limited transmission lengths for active nodes. Additionally, orchestrating multiplexed access requires stricter time synchronization among the nodes.

Three sub-categories of contention-free MAC protocols include frequency division multiple access (FDMA), code division multiple access (CDMA), and time division multiple access (TDMA).

FDMA divides the frequency bandwidth into sub-channels to allow radio pairs to communicate without interfering with other transmissions. This MAC protocol requires channel management to assign sub-channels to stations requesting communication and a multi-frequency radio to change to the channels. In an effort to make the sensor platforms as simple as possible, most sensor transceivers are programmed to only operate on one channel. Therefore, FDMA is not a suitable protocol for sensors.

CDMA operates on a single frequency, but encodes individual transmissions using orthogonal code sets to spread out the power and information of each channel to appear as low-level noise to the other transmissions. CDMA requires expensive hardware and software algorithms to synchronize transmissions and decode the messages.

TDMA scheduling schemes are a major research area for contention-free sensor networks. Using a single frequency channel, TDMA divides the channel into time slots and allocates the transmission slots to network stations. This process is referred to as *scheduling*. Scheduling avoids energy losses due to collisions, idle listening, and message overhearing. Message overhearing occurs when stations receive and discard messages sent to other stations. Figure 2-11 shows a typical implementation of TDMA. The AP uses the traffic control (TC) slot to broadcast upcoming scheduling information. Each participating station is assigned both an uplink and downlink slot in which to communicate with the central coordinator. All other stations may sleep during this time to conserve energy. The final contention period (CP) is allocated to allow nodes to join the network. Reserving time slots for both the uplink and the downlink greatly simplifies the scheduling control overhead [LaH04]. If innovative techniques are developed to implement TDMA in a self-configuring network, TDMA offers the most significant energy-saving opportunities over all other contention and contention-free techniques [LaH04][SoG00][PoK00].

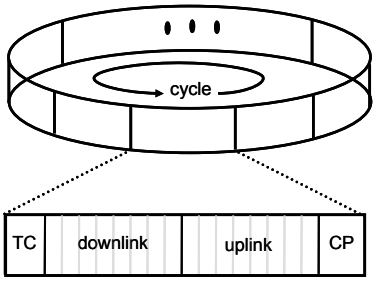

Figure 2-11 TDMA Frame Structure: Traffic Control-downlink-uplink–Contention Period

2.4.3.2 Contention-based Schemes

While schedule-based protocols reduce collisions and increase battery life at the expense of channel utilization, contention-based protocols such as CSMA/CA discussed in the IEEE 802.11 protocol standard sections offer simplicity, scalability, and traffic flexibility. Contention-based WLAN and WSN protocols [WLAN97][ElD04][PoH04] provide mechanisms for competing for access in an asynchronous and random manner. Only stations requiring transmission contend for the bandwidth and can fairly divide up the bandwidth resources. These protocols are

decentralized and work best when devices have bursty, irregular traffic patterns, but they do not perform well under heavy traffic loads since increased frame collision probabilities may lead to network instability. Collisions in wireless sensor networks not only reduce throughput, but their retransmissions also deplete valuable battery resources. Additional energy losses in contention-based protocols include idle listening and message overhearing. Many sensor design implementations try to attain the best attributes from both schedule and contention methods, but must ultimately make design tradeoffs.

2.4.3.3 Slotted Contention-based Schemes

Although many leading researchers conclude that contention-based WSN solutions require too much idle listening and suffer from frame collisions [SoG00][Lah04][PoK00], most of the WSN deployments are built around specialized contention-based protocols called slotted protocols. Schedule-based schemes offer significant energy savings, but they suffer from complexity and bandwidth inefficiency. Slotted protocols offer an alternative scheme to gain the advantages of both schedule- and contention-based protocols [PoH04][YeH04][DaL03]. Slotted schemes resemble the classical contention-based MAC protocols, but they divide the channel into frames with duty cycles to synchronize network functions and allow wireless sensor stations to sleep without having to actively coordinate wakeup times. By synchronizing time into slots that have defined temporal "edges," contention is more organized and stations may take advantage of the network transmission rhythm to exchange data when available or sleep when unavailable. This rhythm of communication provides more opportunities to sleep when the other stations have completed transmissions during the frame period, but increases the probabilities of collision since stations simultaneously begin contending for the medium at the beginning of the next frame period. Synchronizing message transfers and reducing the randomness of the contention period increases the probability of collisions, but most of the slotted protocols add small RTS/CTS control messages to reduce these collision effects. Also, since each station can sleep after all message exchanges, slotted protocols reduce idle listening. Other energy-saving techniques are added to the slotted protocols to reduce the effects of overhearing and the hidden terminal interference.

2.5 Wireless Sensor Network MAC Energy-efficiency Techniques

Wireless sensor network MAC protocols extend network lifetimes by reducing the activity of the highest energy-demanding component of the sensor platform – the radio. Trading off network throughput and latency (delay), energy-efficient MAC protocols synchronize network communication to create opportunities for radios to sleep with active duty cycles as low as 2.5% under minimal traffic conditions [LaH04]. Understanding both normal and malicious sources of energy loss is essential in designing a power control system. Typical sources of energy loss in wireless sensor networks include idle listening, frame collisions, protocol overhead, and message overhearing. This section presents the state-of-the-art WSN MAC protocols designed to optimize energy efficiency.

2.5.1 Idle Listening

Wireless local area network (WLAN) interface devices without energy constraints continuously monitor the medium for incoming transmissions to reduce network response time. Idle listening occurs when a station, or node in the WSN, listens to an inactive medium and dominates power losses in networks characterized by scarce traffic and limited sleep cycles. For example, using the Chipcon CC2420 250kbps transceiver [Chi04], a node can transmit the WPAN maximum-sized 128 byte message in 4.1ms. If a node transmits and receives one 5ms message every second, and then listens to an idle channel for the remaining 990ms, that node spends 99% of its duty cycle expending energy without sending or receiving any messages. Since many wireless sensor radios consume as much or more energy during idle listening than during transmissions, energy-efficient WSN MAC protocols attempt to synchronize network traffic such that transmissions begin only in predetermined time slots. Once all network transmissions are complete for a particular cycle or time *frame*, the protocols allow nodes to return to sleep until the next transmission period. This section presents four techniques to reduce idle listening in WSNs: static sleep scheduling, dynamic sleep scheduling, preamble sampling, and off-line scheduling.

2.5.1.1 Static Sleep Schedule: (SMAC)

Sensor-MAC (SMAC) is contention-based, MAC layer protocol that coordinates sleep periods in a sensor network to conserve energy and increase the network lifetime [YeH02]. This protocol represents the baseline sleep-oriented, energy-efficient WSN MAC protocol design.

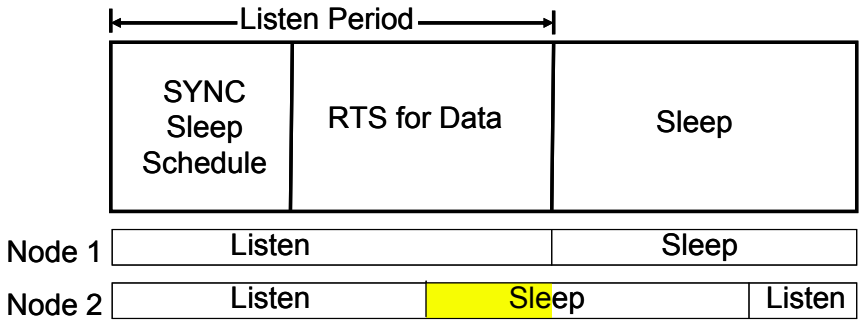

Figure 2-12 SMAC Time Frame Period

SMAC divides every frame into a listen and a sleep period. Figure 2-12 shows that the listen period is further divided into a synchronization period and a data transfer period. The synchronization period allows nodes to periodically announce their sleep schedules to correct network time drift and synchronize their sleep times in order to form a virtual cluster of nodes which have the same active listening and sleep periods. By creating a 10% active duty cycle, node lifetimes can be significantly extended with bounded throughput and latency tradeoffs. Creating a slotted starting time for all network traffic and concentrating the traffic into a smaller time frame reduces idle listening, trading off latency and throughput. The first step in setting a sleep schedule is for a node to listen for a SYNC message from a neighbor. The SYNC message indicates that the sender intends to sleep in *t* seconds. Once the node hears its neighbor's schedule, it adopts the same schedule and retransmits the schedule for other neighboring nodes to adopt. If a node does not hear a SYNC message within a timeout period, the node will set and broadcast its own sleep schedule. Bordering sensor nodes between two active schedules may either adopt both schedules or choose one of them. The bottom portion of Figure 2-12 illustrates the effects of having nodes with misaligned sleep schedules. In this example, if node 1 attempts to transmit to node 2 late in node 1's listening period, node 2 is already in sleep mode and will not be able to receive the message. Unfortunately, SMAC's sleep cycle is fixed at the time of

network deployment. This limitation causes the protocol to be inflexible in responding to network traffic fluctuations or network scaling.

2.5.1.2 Dynamic Sleep Schedule: (TMAC)

Timeout MAC (TMAC) [DaL03] is a contention-based, MAC layer protocol that builds upon the successes of SMAC in optimizing power efficiency for the sensor radio by sleeping during periodic network inactivity. The TMAC protocol introduces a listening timeout mechanism that improves on the idle listening overhead by dynamically adapting the active listening period in response to network traffic. TMAC permits nodes to sleep as soon as all network traffic has completed. As shown in Equation 2-6 and Figure 2-13, the end of traffic is signaled by monitoring an idle channel for an adaptive timeout (TA) period. The TA period represents the longest period in which a hidden node would have to wait before hearing the first bit of a CTS message. This timeout waiting period is decomposed into the largest contention window (t_{CW_Max}), the time to send an RTS message (t_{RTS}), and the protocol small interframe spacing (SIFS) delay before the receiving CTS node can process a response to the RTS.

$$TA = 1.5 * (t_{CW\_Max} + t_{RTS} + t_{SIFS}) \quad (2\text{-}6)$$

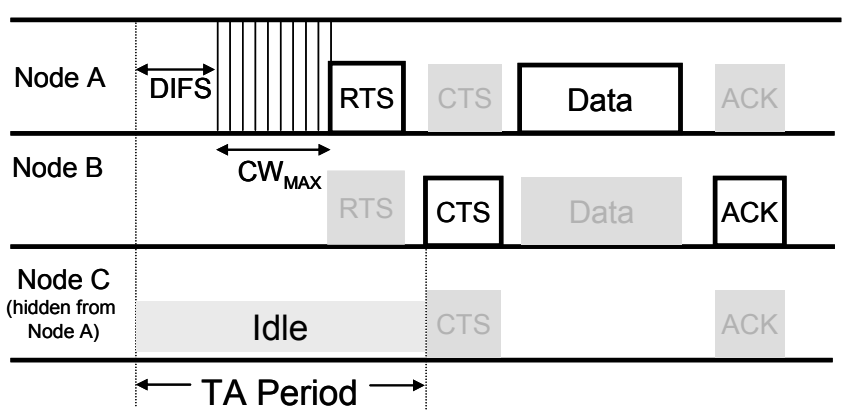

Figure 2-13 Dynamic Sleep Schedule Adaptive Timeout (TA)

Van Dam et al. determined in simulation that a 1.5 scaling factor produced the most stable network [DaL02]. Figure 2-14 contrasts TMAC's adaptive active timeout against SMAC's static

active state. The arrows in the figure indicate bi-directional message traffic and illustrate how TMAC effectively condenses the same number of messages into a smaller time frame to reduce idle listening at the expense of increased message delay. Once a node has waited a timeout period without sensing any traffic, the node transitions to sleep until the next scheduled listening period. In event-based and periodic reporting scenarios, TMAC achieved five times the energy savings as SMAC.

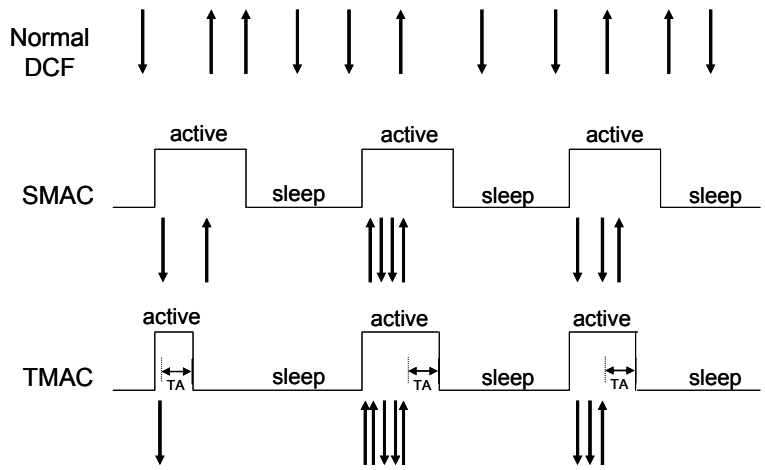

Figure 2-14 SMAC Static and TMAC Dynamic Sleep Periods

2.5.1.3 Preamble Sampling, Low Power Listening, and Low Power Clear Channel Assessment: (BMAC, WiseMAC)

Berkeley-MAC [PoH04] and WiseMAC [ElD04] take a new, decentralized approach to organize sleep schedules by allowing nodes to adopt any sleep schedule, but with a fixed sleeping cycle frequency. At the end of a node's sleep period in both protocols, a node wakes up and samples the channel using a process called low power listening (LPL). The Mica mote WSN platforms manufactured by Crossbow Inc. [Cro06] have the ability to sample the network within 350µs [PoH04]. As illustrated in Figure 2-15, if the node senses activity, it wakes up, synchronizes with an extended packet preamble, and receives the packet. A sender must transmit a preamble length greater than each node's sampling cycle to ensure that the node is awake for synchronization. Since many of the sensor platform transceivers expend more energy receiving than transmitting, experiments have shown that the energy cost penalty for idle listening in the

entire network far exceeds the penalty for transmitting a longer preamble if the system is tuned correctly. In a 40-node network, BMAC predicts that the network can sustain a 2.5% duty cycle. LPL's strategy is to reduce idle listening at the cost of both increased transmitting and receiving costs. This protocol only works well in networks with scarce traffic, and the authors do not take the sleep transition times or energy costs into account.

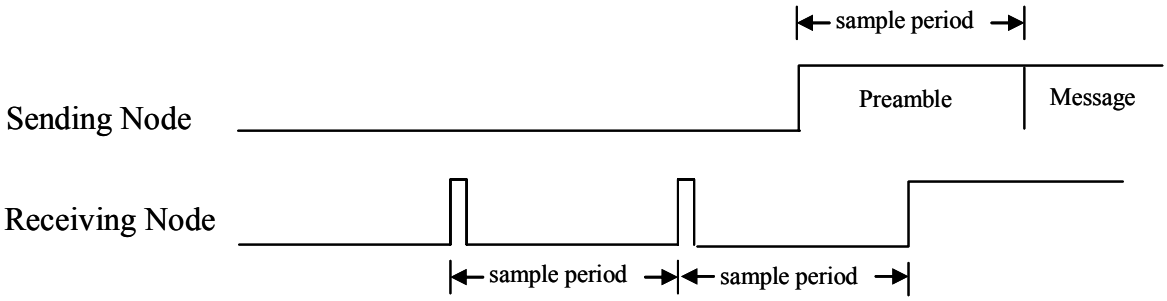

Figure 2-15 BMAC Low Power Listening

In an improvement to BMAC's preamble sampling, nodes using the WiseMAC [ElD04] protocol maintain a schedule of when other nodes wake up. Instead of sending a long preamble, they can send a standard one during the receiver's sensing period. Each node's sleeping schedule information is transferred in the header of other packets to reduce control packet overhead and maintained as a time offset for each node. To account for time drift, the preamble is variably extended according to the duration since the last packet exchange. One limitation to this protocol is the broadcast and multicast messages must span the entire frame.

A deployment experiment using the BMAC protocol [SzM04] found that BMAC's lifetime was reduced between 45% - 75% of the original estimate due to overhearing traffic intended for other nodes. Both BMAC and WiseMAC operate with ultra-low power consumption in networks with low traffic rates.

2.5.1.4 Off-line scheduling: (TRAMA)

Traffic-Adaptive MAC (TRAMA) is a *scheduling-based*, MAC layer protocol that optimizes power savings during inactive periods [RaO03]. TRAMA protocol introduces both a Scheduling Exchange Protocol to schedule message recipients and release unused timeslots for reuse and an Adaptive Election Algorithm to randomly assign transmission timeslots. The

primary goals for TRAMA are to reduce energy requirements by establishing collision-free data transfer and to maximize sleep time. Coupled with these goals are considerations to provide fair access among the nodes, efficient channel utilization, and tolerable latencies.

In general, scheduling-based protocols tend to waste timeslots and bandwidth during periods when scheduled nodes have no traffic to send and other nodes have more traffic to send than slots apportioned. TRAMA's approach alleviates the effects of this limitation by adaptively scheduling nodes based upon their transmission requirements. The TRAMA protocol decomposes into three essential components: the Neighbor Protocol (NP), the Schedule Exchange Protocol (SEP), and the Adaptive Election Algorithm (AEA).

As shown in Figure 2-16, each TRAMA frame is divided into contention and contention-free time periods. The contention period is used to exchange control messages to determine 2-hop neighbors, to synchronize time, and to establish the data transmission requirements. All nodes must monitor the channel during this time period. The contention-free period, on the other hand, is the time for collision-free transmissions of all scheduled packets. Nodes with no activity scheduled may sleep during this period, and nodes with transactions only need to wake up during their scheduled data exchanges. The duty cycle of the contention and contention-free periods is set before deployment according to the application's anticipated mobility support and data throughput requirements. Networks with little or no mobility can allocate more time for the contention-free data transmissions.

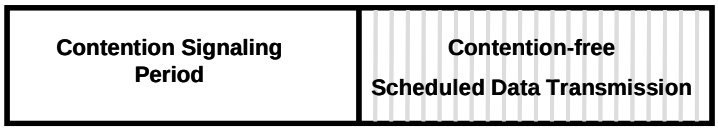

Figure 2-16 TRAMA Frame Structure

The **Neighbor Protocol (NP**) establishes the 2-hop neighborhood for every node and maintains local synchronization. A broadcasted NP control packet sent during the contention signaling period indicates the number of nodes added to and deleted from the sending node's neighbor database and their respective identification numbers. These packets only send incremental changes, so their successful delivery is critical to the development of 2-hop neighborhoods. Since broadcast control messages are too expensive to individually acknowledge, TRAMA relies on theory developed in [BaG02] to establish a 99% probability of

successful transmission. Given N number of 2-hop neighbors, a frame needs to retransmit an individual packet seven times over a retransmission interval T = 1.44* N signal slots. For example, given twenty 2-hop neighbors, each node's neighborhood update should be sent randomly seven times over a period of 29 timeslots. Once every 1-hop node has effectively sent all of its neighbor information, every node will then be able to establish a complete 2-hop neighborhood table. The neighbor exchange messages also serve as local network time synchronization. Sensor network radios transmit extremely limited data rates on the order of 250 kbps [Chi04]; therefore, frame sizes may be on the order of 25msec. With typical clock drifts on the order of microseconds, the networks will automatically resynchronize after every sleep period.

The **Schedule Exchange Protocol (SEP)** allows nodes to advertise the recipients of their queued transmissions, the time slot allocated for each individual packet, and the available timeslots for reuse. SEP saves energy by allowing nodes with no data to send or receive to sleep during the contention-free period after listening to the upcoming schedule. SEP also increases data throughput by reallocating empty timeslots for reuse. Figure 2-17 shows the format for a signaling message sent during the contention signaling period. The sender indicates the number of time periods that the schedule is valid in the timeout field, the width of the node's 1-hop neighborhood using the number of bits in the bitmap field, the number of slots that the node *won* in the Adaptive Election Algorithm (AEA) to send traffic in the slot field, and the individual message recipients for each message listed in the bitmap fields. The signaling message bitmaps

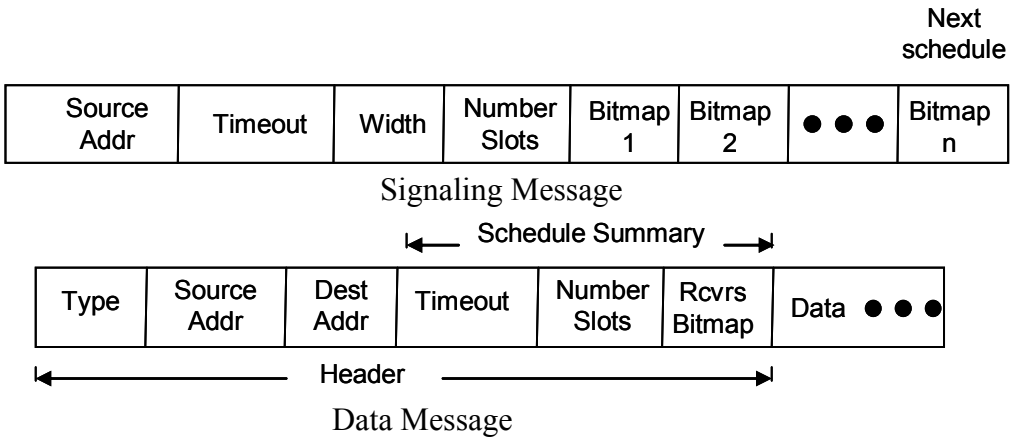

Figure 2-17 TRAMA Signaling and Data Frame Formats

list a bit sequence with a single bit position for each neighbor in the node's 1-hop neighborhood. The bits are listed in decreasing order of magnitude for each neighbor's node identification number. A one is placed in the bit position for each node intended as a receiver. A bit map with all ones indicates that all neighbors would need to be awake to receive a broadcast message. Any subsequent bitmaps containing all zeros are given up for slot reuse. The last bit map indicates the schedule for the node's next signaling/transmission interval. In addition to sending explicit schedule information during the signaling period, a TRAMA data packet header also contains a summary of the schedule as shown in Figure 2-17. The receiver bitmap cues nodes that may not have heard the schedule during the signaling period due to a collision to stay awake for a message queued for them.

The **Adaptive Election Algorithm (AEA)** schedules the transmitters for a given time according to the neighborhood and schedule information exchanged in NP and SEP. A node calculates the time slot priority for each of its contending 2-hop neighboring nodes for every time slot in a data transmission frame. Given the timeslot time (t) and the node's identification number (u), AEA uses a hashing function to calculate node u's priority:

$$\text{priority}(u,t) = \text{hash}(u \text{ concat } t) \tag{2-7}$$

If a node has the highest priority in its 2-hop neighborhood for a given time slot, then the node has the opportunity to transmit data in that time slot or give it up for reuse. In the event that the highest priority node gives up the time slot, the next highest priority node that announced a scheduled requirement will transmit during the time slot. The AEA algorithm also provides a mechanism that allows nodes to account for more than one node within a 2-hop neighborhood of u to consider itself the highest priority node. This situation occurs when the two winning nodes are 3-hops away from each other, yet both are in u's 2-hop neighborhood. Node u would look at the highest priority winner in its 2-hop neighborhood to send traffic, but also consider an alternate winner's schedule for transmission before going to sleep.

TRAMA provides an organized method to configure and maintain a TDMA network, but the protocol uses excessive control packets to exchange schedule information and requires significant processing to calculate the slot priorities. For S time slots and N nodes in the 2-hop neighborhood, every schedule iteration requires each node to process S * N calculations.

2.5.2 Frame Collisions

A frame collision occurs when a wireless sensor node sends a MAC protocol frame, or message, which collides or overlaps in time with another message. If the interfering signal strength is high enough, the data is corrupted at the receiving end. In most single-channel radios, the radio cannot simultaneously receive while in transmit mode. Therefore, the message sender's only indication of a collision is the absence of a message acknowledgement from the receiver. Frame collisions occur naturally in wireless networks due to the extensions of space and time in distributed radio networks. Finite radio receive-to-transmit transition times (the capture effect) ranging from 250 μs to 500 μs after sensing a clear channel; propagation delays between distant stations; and hidden nodes which are out of range of the sender, but within range of the receiver, are the leading causes for wireless frame collisions. Resending messages causes both the sending and receiving node to expend additional energy. Protocol designers reduce frame collisions by employing contention-free scheduling protocols or contention-based backoff algorithms to minimize probabilities of collisions. Typically, link layer parameter settings permit a limited number of retransmissions before discarding the message.

Techniques to reduce or mitigate the effects of frame collisions include the IEEE 802.11 exponential contention backoff discussed in Section 2.4.2, transmission scheduling with TDMA protocols, and 4-way RTS-CTS-data-ACK handshaking to reserve the medium before sending data. Both the SMAC and TMAC protocols use contention and RTS-CTS exchanges to reduce collisions. BMAC offers RTS-CTS as an option available to the application. As shown in the frame control field in Figure 2-18, both the RTS and CTS messages contain a duration field which advertises to all surrounding nodes the duration of the transmission exchange. By having all nodes set their network allocation vector (NAV) countdown timers for the duration of the exchange, the protocol significantly reduces the frame collisions after the initial RTS has seized the channel.

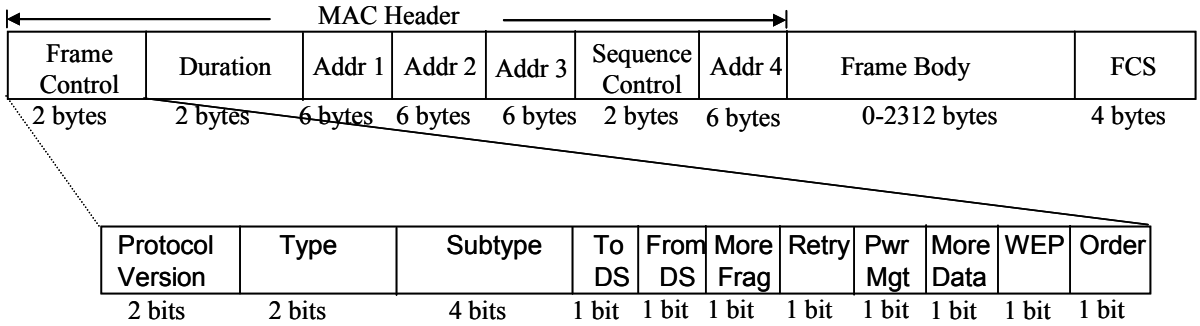

Figure 2-18 IEEE 802.11 Frame Format

2.5.3 Protocol Overhead, Control Packets

Wireless protocol overhead consumes both energy and bandwidth. Networks serve as an integrated system to transfer data between distributed application layer programs, but maintaining a network and providing reliable data delivery requires tradeoffs in effective throughput and energy efficiency. Adding data message headers and 2-to-1 Manchester encoding to the RFM TR1001 [RFM05] transceiver reduces the 115.2 kbps physical transmission rate to an effective 46 kbps [LaH04]. This 60% reduction does not yet include the additional network control required to configure the network using neighbor discovery, synchronize time, and determine available message routes.

Achieving reliable data delivery requires the MAC layer to provide network mechanisms to establish a fair and reliable distribution system. These mechanisms include network protocol overhead for error detection/correction codes, message header source/destination addresses, and medium contention functions. Standard error detection and correction codes double the data size to add redundant data information in each message for corrupted data recovery. Adding the forward error correction (FEC) overhead reduces the application data rate, but prevents costly retransmissions in noisy environments. Section 2.4.2 introduced the IEEE 802.15.4 standard 128-byte packet as having the PHY and MAC layer headers consuming up to 20% of the packet's maximum size. With the 9 to 25 byte frame header, the MAC-layer header control overhead can often be larger than the actual data payload. The WSN DCF contention function adds additional control overhead by requiring nodes to monitor the channel during exponential backoff periods and interframe spaces. These extended sensing intervals cause the receivers to

expend additional energy. WSN scheduling-based protocols conserve the contention energy, but they expend more energy and bandwidth exchanging scheduling messages. Finally, using the RTS-CTS-data-ACK CSMA/CA protocol to reserve, transmit, and acknowledge a message adds more control overhead.

2.5.4 Message Overhearing

Receiving and discarding messages intended for other nodes, or message overhearing, is a common procedure in non-energy constrained networks. Receiving all messages is an efficient method to increase throughput and decrease latency, but it also causes all of the receiving nodes to expend energy, especially in cases where the radio receive mode expends more energy than the transmission mode. Two energy-efficient techniques exist to reduce message overhearing: early rejection and message passing. Early rejection allows a sensor node to turn off its radio once it has read the destination field for an incoming unicast message or the group id for a broadcast message and determined that it is not a receiving node [KaS04]. Message passing is implemented in SMAC and TMAC. This technique allows nodes to schedule a sleep period during an overheard RTS-CTS handshake sequence by noting the message duration field and scheduling a network allocation vector (NAV) table interrupt [KaS04][YeH02][DaL03].

Message overhearing occurs when a node actively receives messages transmitted over the shared medium which are not destined for them. Table 2-2 illustrates how receiving a message consumes three to four orders of magnitude more energy than the powered-down mode.

Table 2-2 Receive and Sleep-mode Current Consumption

Radio	Receive mode	Power-down mode
CC2420 [Chi05]	19.7 mA	1 μA
CC1000 [Chi05]	9.6 mA	0.2 μA
RFM TR1001 [RFM05]	3.8 mA	0.7 μA

SMAC employs message passing to reduce energy consumption and latency by parsing long messages into smaller fragments. Wireless channel characteristics introduce the uncertainty of reliable data transmissions with variable probabilities of induced bit errors over time. To reduce the probability of costly retransmissions and added latency, SMAC uses RTS-CTS to gain medium access, and then transmits a burst of fragments of the larger message. As shown in

Figure 2-19, the receiver responds with an acknowledgement (ACK) message after each successful fragment transmission. Unlike IEEE 802.11 fragmentation, the duration fields in the message passing RTS-CTS messages account for **all** of the fragments, the ACK messages, and the associated short interframe (SIF) spacing periods. Although retaining control of the medium for the duration of all of the fragments may appear unfair, the idea that the sensor nodes are working together toward a common application creates an application-level fairness by decreasing the end-to-end latency of the entire message.

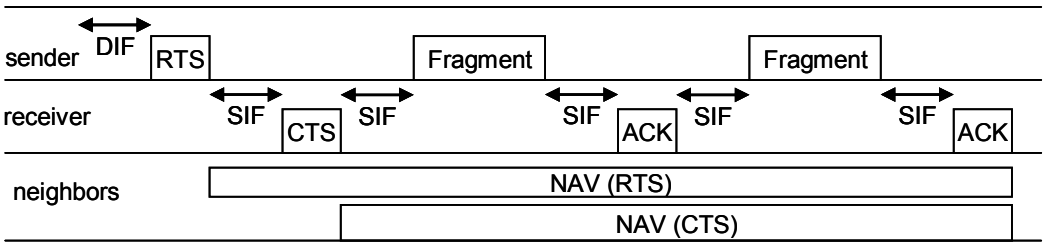

Figure 2-19 Message Passing Timing and Signaling

2.6 Additional TDMA Sensor MAC Energy-efficient Protocols

The WSN energy-efficient MAC protocols discussed in the previous section represent the state-of-the-art contention-based protocols. This section presents cluster-oriented time division multiple access (TDMA) MAC protocols to illustrate the potential energy savings attained by these synchronized schemes.

2.6.1 Low-energy Adaptive Clustering Hierarchy (LEACH)

Low-energy adaptive clustering hierarchy (LEACH) [HeC00] is a self-organizing, cluster-based protocol which uses a passive mechanism to randomly select a cluster head node. LEACH assumes that all nodes are homogeneous and are deployed at the same time with equal energy. LEACH's key features are energy-saving data aggregation/fusion to reduce the amount of data messages sent back to the base station, localized coordination for cluster setup and operation, and randomized rotation of the local base station or cluster head nodes. This innovative "off-line" cluster head election uses a probability-based algorithm to passively choose the next round's cluster head without passing any control messages. Once the self-election is complete, the

cluster head uses CSMA to broadcast a cluster advertisement message to all of the local nodes within radio range. The other nodes listen to all of the cluster head broadcasts and attempt to join the cluster head with the highest receive signal strength indication (RSSI). This cluster will require the least amount of energy for intercommunication. The cluster head gathers the cluster membership requests and builds a schedule for each node to send its data up to the cluster head to be aggregated and forwarded to the network sink. This traffic pattern only provides for data flowing out of the network, not allowing nodes to communicate with one another.

The offline cluster head election process indirectly uses each node's remaining energy level to determine the node's probability of self-election. The system parameter, P, set before deployment determines the optimal percentage of simultaneous cluster heads to have in the network.

$$T(n) = \frac{P}{1 - P*(r \bmod \frac{1}{P})} \; if \; n \in G, 0 \; otherwise \qquad (2\text{-}8)$$

The parameter r represents the current election round, and G represents the nodes which have not been a cluster head in the past $1/P$ rounds. For example, if $P = 0.05$ (5% of network cluster heads per round), each self-elected node will not be eligible again for 1/0.05 = 20 rounds. If the number of simultaneous clusters is too low, many nodes will have to transmit their data farther than is efficient. Too many cluster heads places more of a burden on group to transmit farther to the base station. Although this probabilistic election method statistically spreads the rotations of the cluster head nodes, the algorithm does not take the actual energy level of the node into consideration. Even in an initial homogeneous network, nodes which process more sensor events may consume energy at a higher rate. Requiring them to serve an equal number of cluster head rotations will reduce the overall network lifetime

2.6.2 Power Aware Clustered TDMA (PACT)

Power aware clustered TDMA (PACT) [PeC01] is a protocol which adapts its duty cycle to the user traffic. Like LEACH, PACT uses a passive cluster head election scheme, but PACT bases the election eligibility on the node's battery energy level. These cluster heads then form the communication system backbone nodes. Experiments have shown that clustering improves ad hoc on-demand vector routing (AODV) by only permitting cluster and gateway nodes to forward route requests, thus reducing the number of flood messages [GeK00].

PACT classifies nodes into four status categories: cluster head, gateway, ordinary, and low energy state (LES) node. Each node exchanges its current status as a two-bit field appended to every message. An existing cluster head will broadcast an LES status when its energy level falls below a certain threshold. The departing cluster head node will then only be able to participate in sensing and collaboration unless the node is recharged. Nodes in the low energy state will be ineligible to be a cluster head. In an attempt to become the next cluster head, all eligible nodes will add the cluster head code to their next message [GeK00]. The first node claiming to be the new cluster head assumes the responsibility.

PACT uses the TDMA superframe shown in Figure 2-20. Each superframe is composed of control mini-slots where all nodes broadcast the destination addresses for the follow-on data slots. They also use these messages to broadcast their current cluster role status.

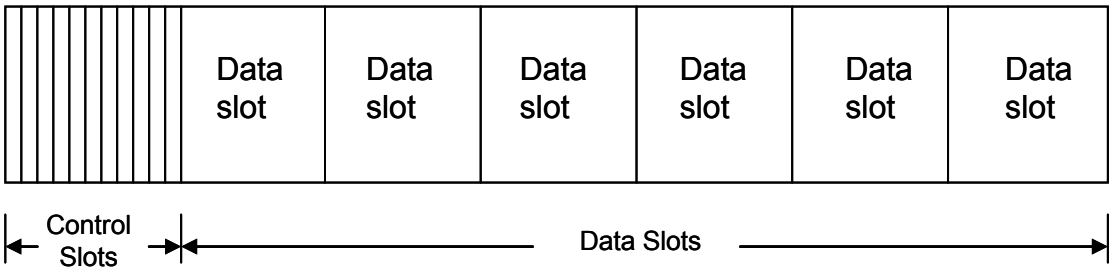

Figure 2-20 PACT TDMA Superframe

Unlike many TDMA schemes, PACT provides for both forwarding messages out of the cluster *and* passing messages within a cluster. On the other hand, although nodes can turn off their transceivers during data slots which are not intended for them to receive or transmit, each node must be awake for *n* number of control slot messages to determine the follow-on schedule, where *n* is the number of nodes in a cluster.

2.6.3 Bit-Map-Assisted Energy-efficient MAC Scheme for WSN (BMA)

The intra-cluster communication bit-map-assisted (BMA) MAC protocol [LiL04] is designed to provide a one-way communications path from cluster members to a cluster head in event-driven wireless sensor networks. The BMA protocol's major objectives are to reduce energy losses caused by idle listening and collisions while maintaining low latency transactions.

BMA creates network clusters in the same manner as LEACH. Nodes self-elect to become cluster heads based upon the decentralized probability equation. After all of the elected cluster heads have broadcast their member advertisements, nodes join clusters based upon the highest receive signal strength indication (RSSI). After forming clusters in the setup phase, Figure 2-21 illustrates the nodes entering the steady-state phase in which each transmission's successive session consists of a contention period, a data transmission period, and an idle period.

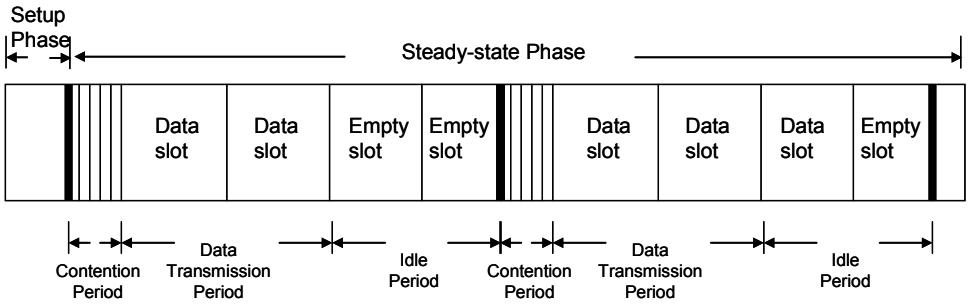

Figure 2-21 Bit-Map-Assisted Frame Structure

The contention period allows the cluster members to communicate to the cluster head of their transmission requirements. Each node is allocated a "1-bit" control slot in the contention period to transmit its need for a data slot. If the node does not require a data slot, the slot remains empty. At the end of the contention period, the cluster head announces the schedule for transmission. Cluster members sleep during all data slots except for their own if they need to transmit. After all of the data slots are complete, the session enters the idle period to allow all cluster member nodes to sleep.

The BMA offers nodes the ability to sleep for long durations during the contention-free data transmission and idle periods. The ability to synchronize a wireless sensor network and communicate a *1-bit* message is an unrealistic assumption. The state-of-the-art radios transmit 4-bit symbols in 16μs [Chi05]. The protocol performs well in the analytic models, but a remote area deployment will suffer synchronization problems. Additionally, having all nodes active during the contention period causes unnecessary message overhearing.

2.7 Wireless Sensor Network Security

Besides the classical denial of service attacks that plague the IEEE 802.3 wired and IEEE 802.11 wireless networks, WSN networks have a unique vulnerability due to their fixed energy capacity upon deployment. The mote systems generally operate on a limited energy capacity. Malicious attackers can easily target the battery supplies and reduce network lifetimes from years to days. Achieving a secure system requires security integration into every component to prevent a vulnerable point of attack [PeS04][PoH04]. WSN designers must incorporate protecting the critical energy resource into the system architectures.

Authentication and encryption solutions for the resource-constrained WSN cannot achieve the same protection levels as wired and wireless infrastructure-based networks. Securing communications in WSNs entails a secure group approach to support nodes performing in-network processing and aggregation necessitated by wireless bandwidth limitations [PeS04]. Additionally, individual node's limited key storage capacity prevents the use of per-link encryption strategies. This section introduces two leading software and hardware tools to maintain group security and performance for WSNs.

2.7.1 TinySec

TinySec is a software-based link layer security architecture which provides the basic necessities in network security, authentication, and encryption [KaS04]. The network link layer is the logical choice for implementing these security components in sensor networks since the sensor traffic tends to involve either source-to-sink or broadcast traffic (one-to-many), not end-to-end (one-to-one) traffic like traditional networks. TinySec is a module which pairs with the TinyOS operating system to provide a lean, component-based operating system to manage the sensor platforms. The design goals for this implementation are to provide link layer security measures to protect access control, message integrity, and message confidentiality without significantly impacting the energy and throughput of the network. TinySec pairs a two-byte counter and source field with the traditional TinyOS packet header fields (destination, active mode, length) to develop an initialization vector (IV). Since the IV uses integral header fields, the encryption mechanism minimizes control overhead. TinySec pairs the 8-byte IV with a cyclic block code (CBC), Skipjack, to attain reasonable security without a computationally

complex algorithm. Experiments using Advanced Encryption Standard (AES) and Triple-Data Encryption Standard (DES) proved to be too slow without hardware acceleration. Additionally, TinySec replaces the medium access control frame's cyclic redundancy code (CRC) with the message authentication codes to maintain the 8-byte security control overhead. To allow the sensor application the ability to tradeoff security to reduce computation and communication overhead, TinySec offers the ability to operate in three modes: open, authentication (TinySec-Auth), and authentication-encryption (TinySec-AE).

2.7.2 Hardware Advanced Encryption System (AES)

Offering a hardware alternative to TinySec's software components, IEEE 802.15.4-compliant radios provide the ability to perform 128-bit AES hardware encryption on the radio platform. The AES is a Federal Information Processing Standard (FIPS) which provides a cryptographic algorithm to protect electronic data by implementing a symmetric block cipher to encrypt and decrypt information [Nit01]. One example is the Chipcon CC2420 [Chi04] which provides four security modes:

- Disabled
- Cipher block chaining (CBC-MAC) authentication
- Counter (CTR) encryption / decryption
- Counter with Cipher Block Chaining-Message Authentication Code (CCM) authentication and encryption / decryption.

As with TinySec, the integrated 802.15.4 [WPAN03] security mechanism provides message integrity and confidentiality. Incorporating these algorithms using hardware and software routines located directly on the radio releases the processor's limited memory and processing capacity to handle other operations.

2.8 Wireless Sensor Network Timing Considerations

Wireless sensor networks require synchronized timing among nodes in order to fuse sensor data for coordinated analysis, to timestamp events, and to coordinate slotted or TDMA MAC protocol schedules. Message time delays are attributed to send processing, channel access delay, transmission delay, medium propagation delay, reception delay, and receive processing delay. Synchronization error in wireless networks primarily results from the non-deterministic random

time delays between nodes [PaS04]. The Internet's established Network Time Protocol (NTP) [Mil94] performs well in large multi-hop network environments built upon a hierarchy of time servers and routers. Depending on their application, resource-constrained WSN networks may require even more precision than the Internet with significantly less bandwidth, energy, and computational power available to devote to the time synchronization process [ElG02]. Since WSN networks are deployed in environments that may not be able to access GPS satellite time to maintain absolute time, many WSN rely solely on relative time synchronization to determine the order of events or perform relative measurements among nodes. Synchronizing WSN nodes in an efficient manner requires selecting an approach which leverages as much of the existing communications transactions as possible to minimize control overhead. One efficient method is to couple the application and the MAC layers by applying a timestamp onto a message immediately before transmission. A MAC layer timestamp mitigates the most significant point of uncertainty in packet delay, random channel access delay. Hill et al. contend that MAC layer timestamps with the Mica mote can obtain microsecond accuracy on the timestamps [HiC01] since the limited range in WSN networks makes the propagation delay negligible. Subsequent research shows that even using the MAC layer timestamp, only an average of 20µs synchronization accuracy is currently attainable [GaK03], with a 50µs worst case. Another approach that does not require cross-layer communication is the Reference Broadcast Synchronization (RBS)[ElG02]. RBS relies on broadcasted messages to allow receivers to build relative offset tables for the nodes in their neighborhood. With a small synchronization overhead, RBS was able to achieve 12µs synchronization accuracy, eight times better than the Internet's NTP.

An important design tradeoff issue is trading synchronization accuracy for energy efficiency. Ganeriwal et al. claim to obtain five times the accuracy of RBS with the Time-synch Protocol for Sensor Networks (TPSN) [GaK03]. This protocol performs two pair-wise synchronization message exchanges to calculate the clock drift and propagation delay between the two nodes. They present an equation based upon the specified clock drift, T_{drift}, and the worst case synchronization error for a given protocol, $T_{syncError}$, to determine the required synchronization period ($T_{synchPeriod}$) to achieve a target synchronization accuracy bound ($T_{accuracy}$).

$$T_{accuracy} = T_{synchError} + (T_{drift} * T_{synchPeriod})$$
$$T_{synchPeriod} = \frac{T_{accuracy} - T_{synchError}}{T_{drift}} \qquad \text{(2-9 and 2-10)}$$

Using the experimental data from the TSPN protocol [GaK03], the synchronization period required to attain a 500µs accuracy requires:

$$T_{synchPeriod} = \frac{T_{accuracy} - T_{synchError}}{T_{drift}} = \frac{500\mu s - 50\mu s}{4.75\mu s} = 94.7 \text{ seconds} \qquad \text{(2-11)}$$

Additionally, a one minute synchronization period would produce 335µs synchronization accuracy. Although decreasing the synchronization period can have the accuracy approach 50µs, the energy and throughput costs to transmit the synchronization messages limits precludes attaining this level of accuracy.

2.9 Wireless Sensor Platforms

In an effort to make inexpensive sensor platforms ubiquitous, IEEE 802.15.4 LR-WPAN devices have limited processing capability, memory capacity, and battery life. Small system platforms which integrate sensors, processors, and transceivers are referred to as motes. The architecture has four major components: the processor, the transceiver, the sensor interface, and the secondary memory. Table 2-3 illustrates the power and memory limitations for four leading motes. The first mote, Smartdust, is the result of a UC Berkeley/DARPA research project which designed and built a 1mm^3 WSN platform [KaK99]. This device represents the extreme limit for WSN resource constraints. The 128kB and 512kB EEPROMs on the other three motes also significantly limit the code size available to implement sophisticated protocols. Developers implement the sensor protocols in NesC, a structural component-based programming language, and link it to the hardware through a lean, open-source operating system – TinyOS [LeM04] [TOS05]. As technology continues to advance, the wireless sensor trend will continue building physically smaller and more energy-efficient platforms.

Table 2-3 Mote Microcontroller / Transceiver Platform Specifications

Platform	Smartdust [KaK99]	MICA2 [Cro05]	MICAz [Cro05]	Tmote Sky [Mot05]
Microcontroller	8-bit	16-bit ATMega 128L	16-bit ATMega 128L	16-bitTI MSP430
MCU RAM	512B	4kB	4kB	10kB
EEPROM/Flash	512B	128kB	128kB	48kB 1MB(external)
Radio	916MHz	868MHz CC1000 [Chi04]	2.4GHz CC2420 [Chi04]	2.4GHz CC2420
Data Rate	10kbps	76.8kbs	250kbs	250kbs

2.10 Sensor Traffic Modeling

Since wireless sensor networks deliver event-driven (scheduled or unscheduled) data to an application for further processing, their traffic does not have the same multi-hop, random behavior as other ad hoc networks. In many applications the generated data may be shared locally to preprocess the data, but then it is sent to a sink node to deliver the data to the application. The data flows in the same general direction, and traffic around sinks tends to be higher [DaK03]. Since sensor networks are designed to support specific data collection applications, their traffic patterns also depend upon the format of the information collected (raw data, pictures) and the frequency of the collections. In general, sensor network traffic patterns fall into the two categories: convergecast and local gossip [KuA04].

2.10.1 Convergecast

Convergecast traffic originates in a sensor node and travels across a sensor network for collection by a sink node for delivery to an application. Sink nodes usually have additional resources such as memory, power, or extended reach-back communications in order to collect the data at the remote location and transmit the data to the application for further processing.

Challenges for networks which use convergecast traffic include configuring the network and developing efficient routing routines. Intermediate routing sensor nodes expend an excessive amount of their energy resources forwarding traffic to the sink node, thus reducing the network lifetime.

2.10.2 Local Gossip

Local Gossip is event-driven traffic which originates in a sensor node and travels to other sensors in order to collaborate or aggregate data before sending this preprocessed data on to a sink using convergecast [KuA04]. Local gossip applications may include seismic sensors comparing data readings to determine the size and direction of an object or ground wave moving through the area or bridge sensors calculating the effects of bridge movement at various points as vehicles pass. The advantage of local gossip is to reduce the amount of traffic that flows throughout the sensor network, thus saving energy. Redundant information that is observed by multiple nodes may be discarded or raw data may be refined into a more condensed data package. The cost of preprocessing the information is increased sensor platform processing complexity and possibly more local transmissions to exchange information and then pass the data out of the network. In the case where the sink node is nearby the sensors and the reach-back link is not resource-constrained, many sensor network designers prefer to send the raw data in a convergecast manner back to the application for processing.

2.11 Summary

This chapter presented the basic theoretical background and a review of related research in the area of wireless sensor network energy-efficient MAC protocols. In order to understand the nature of wireless networks, the chapter began by explaining the challenges encountered in both the wireless local area networks and wireless sensor networks.

Following the characterization of the wireless medium, highlights from each layer of the network protocol stack provided the framework for the interfaces and functions required from the medium access layer. Next, the current wireless local area network (WLAN) and wireless personal area network (WPAN) standards presented the foundational mechanisms used in commercial network implementations. The succeeding wireless sensor network (WSN)

examples showed how research techniques improved on these standards in an attempt to reduce the primary sources of energy loss: idle listening, frame collisions, protocol overhead, and message overhearing. These WSN protocols improved upon the energy-saving capabilities of the traditional WLAN approaches, but the SMAC and TMAC DCF-based protocols continued to require network-wide idle listening. The SMAC protocol's fixed duty cycle was unable to respond to energy-saving opportunities in low-traffic conditions or to expand the bandwidth in high-traffic conditions. TMAC added a dynamic duty cycle to overcome the limitations of SMAC, but the adaptive timeout period provided a fixed, network-wide idle listening cost for every frame. This energy overhead significantly reduced the TMAC network lifetime under all traffic conditions. Additionally, all of the schedule-based approaches analyzed in this chapter produce excessive control overhead.

Wireless sensor network security and network timing were presented next to provide architectural completeness to MAC layer design. Then, the current wireless sensor transceiver devices were introduced to provide a group of target research platforms. Finally, the nature of wireless sensor network traffic concluded the chapter to illustrate the necessity of bidirectional traffic within a wireless sensor network cluster.

This research is motivated by the need to further reduce WSN energy consumption. The following chapter details the methodology and approach for designing an energy-efficient WSN protocol that extends the work of previous research by creating a cluster-based network which not only reduces, but eliminates the network-wide effects of idle listening, frame collisions, and message overhearing.

Chapter 3

Objectives and Methodology

Learn as if you'll live forever,
Live as if you'll die tomorrow.
--Gandhi

This chapter presents the objectives and methodology used throughout this research. The purpose of this research effort is to develop a network MAC layer protocol which extends the network lifetime for wireless sensor networks. The key contributions of this research are the design and implementation of a centralized WSN network cluster management system which increases the energy savings in existing WSN MAC protocols and of a radio power management (RPM) algorithm designed to increase the energy-saving opportunities of the newest generation of WSN platform transceivers. This work also introduces and validates an innovative technique to collect and deliver sensor network data along the highway.

3.1 Ten-Step Performance Evaluation Method

Selecting an appropriate, proven methodology is a critical step in any research endeavor. Whenever possible, the actual WLAN and WSN equipment was used to implement and test the designs presented in this research. Both analytical and simulation models were used to validate

the energy-saving capabilities of the WSN MAC protocol. The simulation model was developed using the *Jain ten-step method* for a systematic approach to performance evaluation. The design sequence outlined below is well suited for evaluating the performance of a communications system through simulation and testing [Jai91]. The design steps are:

1. State goals and define the system
2. List services and outcomes
3. Select metrics
4. List parameters
5. Select factors to study
6. Select evaluation technique
7. Select workload
8. Design experiments
9. Analyze and interpret data
10. Present results

3.2 Goals and Defining the System

Chapter 2 presented the necessary background to understand the extensive energy-related challenges faced in wireless sensor networks and introduced both standard- and research-oriented techniques designed to reduce their effects. The primary goal of this research was to develop a new, innovative WSN MAC protocol called Gateway MAC (GMAC). GMAC extends the state-of-the-art, energy-saving techniques by focusing on the establishment of a traffic rhythm to maximize sleep duration opportunities, minimize control traffic overhead, and increase network lifetime. TMAC provided an innovative method to reduce idle listening in a DCF-based network, but the cost of the network-wide idle listening for an adaptive timeout period consumes valuable energy and reduces the network lifetime. The motivation for GMAC was to develop a protocol which eliminated network-wide idle listening. Additionally, this research developed a radio power management (RPM) algorithm to enhance existing energy-efficient MAC protocols by studying the new generation of WSN physical transceiver platforms and characterizing their sleep transition capabilities. The final goal of this research was to

validate an innovative wireless distribution system concept called the symbiotic wireless distribution network.

3.2.1 Network Under Investigation

Defining the system requires delineating the boundaries of the system being tested and stating the necessary assumptions. This research focuses on MAC protocol energy-saving performance. The network under study is fully connected with all of the nodes within communications range of one another. The MAC protocol's modular architecture supports any proactive or reactive routing protocol without requiring any modifications.

The application sensor network class that this research augments is remotely deployed in an area which does not provide the capability to physically monitor or maintain the sensors. The sensor message traffic patterns consists of both the convergecast and local gossip models discussed in Section 2.10, requiring both inter-network and intra-network communications. Each of the nodes is deployed with a fixed energy source. Their sensing applications may be different, but their processors and communications platforms are identical.

To delineate design boundaries to focus on the MAC layer energy-efficiency mechanisms, the design makes the following assumptions:

1. Homogeneous, resource-constrained nodes: Identical nodes provide a simpler energy consumption model to analyze the network lifetime.
2. No real-time latency constraint: The strategy of most energy-saving MAC protocols trades latency for energy efficiency.
3. Single-hop sensor network: The gateway sensor network specifically targets a single-hop sensor network. To support more general solutions in which this network represents a cluster or bordering network within a multi-hop sensor field, this modular architecture may be extended to support any proactive or reactive routing protocol without any MAC layer modifications.
4. Fixed power: The sensor platforms are deployed with a fixed, DC power supply. Although solar or battery supplies may be deployed to extend network lifetime, this assumption simplifies network lifetime analysis and comparisons.

5. Battery voltage level can be measured accurately: Most of the new generation transceivers have a built-in, programmable threshold function to provide an interrupt when the battery falls below a specific setting. This threshold can be reprogrammed while in operation to incrementally monitor the declining battery voltage.

Additionally, the nature of the ultra-low power wireless sensor networks requires that the packet rate and packets size be constrained to provide extended network lifetime.

3.2.2 MAC and PHY Layer Implementations

The WSN research uses the 802.15.4-based WSN network platform processor, memory, power, and radio characteristics to facilitate the acceptance of the new MAC protocol within the WSN community and promote rapid field deployment. All aspects of the GMAC protocol and RPM Algorithm design and testing meet the protocol requirements of the IEEE 802.15.4 low rate - wireless personal area network (LR-WPAN) [WPAN03] standard and the energy and response characteristics of the Moteiv Tmote Sky and Crossbow MICAz sensor platforms [Mot06][Cro06]. Specific system parameters used for protocol modeling and comparison are described in Section 3.6.

3.3 Evaluation Techniques

The selection of a particular evaluation technique can significantly impact the outcome of a performance evaluation. Three possible techniques of performance evaluation are analytical modeling, simulation, and measurement [Jai91]. These methods differ in terms of accuracy, cost, and required time. Measurement is a feasible evaluation method to characterize the WSN radio platforms for the RPM algorithm and validate the symbiotic network interfaces at highway speeds. The IEEE 802.11 WLAN and IEEE 802.15.4 Low-Rate WPAN equipment is available. The results of these measurements were used in the GMAC OPNET simulations. Due to time limitations, programming, debugging, and measuring an actual wireless sensor network are beyond the scope of this research. Analytical solutions typically offer less accuracy than simulation, but are less costly and time consuming. Analytical models are used to verify the simulation results.

3.4 GMAC and RPM Algorithm Simulation Scenarios

The simulation was designed employing a top-down approach using OPNET Modeler 11.0. The hierarchical structure of network scenarios, nodes, and processes provides a comprehensive developmental environment to model WLAN and WPAN networks. The Discrete Event Simulation (DES) tool in the OPNET Modeler has a fully developed IEEE 802.11 wireless package to modify in order to model both the behavior and performance of GMAC.

The GMAC simulation evaluation involved implementing GMAC, SMAC, and TMAC in OPNET and using an object-oriented approach to vary packet, network, and traffic sizes to analyze their performance. The protocols without the RPM algorithm evaluation initially benchmarked each of the standard models developed in the GMAC scenarios using strict power management mechanisms which only allow nodes to power off if they have sufficient time. Follow-on simulations incorporated the RPM algorithm under the same conditions and measured the additional energy savings. The purpose of the GMAC and RPM algorithm simulations was to sufficiently address each of the research questions posed in Section 1.4.

3.5 Simulation Performance Metrics

The performance of the WSN MAC protocols was evaluated based upon network lifetime, energy consumption per bit, throughput, end-to-end delay, and node sleep percentage. These performance metrics are defined as follows:

3.5.1 Network Lifetime

Network lifetime is a measurement that can be categorized as either the time from network deployment to the first node failure or the time when the wireless sensor network connectivity becomes partitioned. This measurement provides a fair evaluation of how all nodes work together as a network system to extend network longevity. The GMAC and RPM algorithm performance evaluations measured the time from network deployment until the failure of the first node. Network lifetime is expressed in days, and the performance rating increases with a higher number of days.

3.5.2 Energy Consumption/bit

Energy consumption/bit is a measurement of how much total energy it takes to deliver the network data bits. In addition to transmitting and receiving the data bits, protocols also expend energy performing functions such as processing network information, exchanging control messages, and listening to an idle channel. The energy/bit performance measurement compares protocol energy efficiency to the data traffic loads. Energy consumption/bit measures the average energy consumed per bit in a successful packet transaction in joules / bit, and the performance rating increases with a lower energy consumption/bit.

3.5.3 Throughput

Throughput is a measurement of how quickly data information flows across the channel or network. Throughput has physical limits, and protocol overhead, data encoding schemes, error detection and correction processes, or message retransmissions can slow down the data flow. Throughput was measured in bits/second and packets/second, and the performance rating increases with higher rates.

3.5.4 End-to-End Delay

End-to-end delay is a measurement of the network delay on a packet and is measured by the time interval between when a message is queued for transmission at the physical layer until the last bit is received at the receiving node. OPNET measures this period as the average elapsed time between source transmission and sink reception. The end-to-end delay was measured in seconds, and the performance rating decreases with increasing time.

3.5.5 Node Sleep Percentage

Node sleep percentage is a measurement of the amount of time a node spends in a reduced power sleep state throughout the network lifetime. The sleep percentage is measured by accumulating the amount of time a node spends in any of the low power sleep modes and dividing that time by the node lifetime. The sleep time includes the sleep state transition and recovery times. The node sleep percentage is measured as a ratio of sleep to total node lifetime, and the performance rating increases with increasing percentage.

3.6 Simulation Factors

Parameters which are varied during an experiment or simulation to study their effect on the system performance are called factors [Jai91]. Factors that were varied in the simulation include the MAC protocol, the number of nodes in a network, the radio power management algorithm, and the network aggregate packet rate in order to study the network lifetime, energy/bit, throughput, network delay, and sleep percentage.

3.7 System Model

For the simulation to represent the class of ultra-low power LR-WPAN wireless sensor networks, the system parameters must be properly determined to provide a valid model. This section analyzes each of the primary system model parameters to achieve a fair and representative model.

3.7.1 Network Topology

The network topology can have a significant effect in WSN networks when establishing a single-hop network cluster. The IEEE 802.15.4 class of wireless sensor platforms is typically limited to an outdoor range of 125m [Mot06]. The WSN topologies used in the OPNET Modeler simulations randomly place 50 nodes in a 100m by 100m area.

3.7.2 Number of Nodes

WSN applications may have tens to thousands of sensors in a network. The data frame formats in the simulation models reserve an address space for a single-hop network of 254 nodes, with two logical addresses reserved for broadcast messages and the GMAC gateway. Initial analysis showed that GMAC can gain significant network lifetime over the other WSN protocols under study with as few as 5 nodes. Given the limited 100m x 100m area (10 km^2) for a single-hop cluster, 50 nodes was chosen as a reasonable cluster sample size.

3.7.3 PHY Layer Model

The OPNET Modeler PHY pipeline stages represent all relevant transmission link considerations. These stages include transmission delay, propagation delay, link closure,

modulation techniques, and bit error rate (BER). Due to the close proximity of the network topology, only the PHY transmission delay and associated radio energy costs were included in the MATLAB analytical model.

3.7.4 Data Rate and Slot Time

The IEEE 802.15.4 LR-WPAN standard limits the data transmission rate to 250 kbs. Modeling the simulation and analytical model after the IEEE 802.15.4-compliant Chipcon CC2420 radio, the data rate was set to 250kbs. This radio uses a direct sequence spread spectrum baseband modem with 2 Mchips/sec. The modulation format is Offset – Quadrature Phase Shift Keying (O-QPSK) and transmits one 4-bit symbol in a 16μs time slot. The resulting byte rate is 1 byte per 32 μs, or two symbol timeslots.

3.7.5 Packet Generation Rates

Zheng [ZhR05] considered 100 bytes/s a heavy traffic load during the development of the PMAC WSN energy-efficient protocol. In 2003, UC Berkeley [SzM04] [SzP04][MaP02] conducted an environmental and habitat "sense and send" experiment over a four-month period on Great Duck Island, Maine. The experiment measured infrared heat from bird nests, temperature, barometric pressure, and humidity. The requirement was to determine reliable sample readings every hour. Oversampling in uncertain network reliability conditions produced a sampling rate of 5 to 20 minutes per sample. The single-hop network under study consisted of a two-tiered architecture with a resource-rich gateway and 49 sensor motes. This rate produced an equivalent of 6 to 24 seconds per packet. This research analyzes packet rates extending from 0 to 60 packets/s (or infinity to 16.67 ms per packet) to extend the range of GMAC supported applications.

3.7.6 Data Packet Sizes

The IEEE 802.15.4 [WPAN03] standard implicitly sets the maximum supported MSDU packet size to 128 bytes. As shown in Figure 3-1, the physical header uses a 1 byte *frame length* field to specify the MSDU length. Reserving the most significant bit (MSB) for future use, this 7-bit length represents 0-127 sizes. The Chipcon CC2420 implementation of this standard provides a 128-byte buffer to store an MSDU for transmission. The UCB Great Duck [SzM04]

experiment used 64-byte messages to communicate sensor readings to a network sink node. This research model generalizes the application to send data in a uniform distribution from 32 bytes to the maximum possible of 117 bytes. The 11-byte WSN MSDU header used for all protocol comparisons increases these MSDU data sizes to a range of 43 bytes to 128 bytes.

Octets: 4	**1**	**1**		**variable**
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

Figure 3-1 IEEE 802.15.4 PHY Header [WPAN03]

3.7.7 MAC Frame Period Size

Choosing a MAC frame period size requires making design tradeoffs for sleep duration opportunities, network response time and network synchronization time drift. Analysis developed in the next chapter reveals the ability of GMAC to maintain SIFS duration network synchronization with frame times as long as 31 seconds. Shorter frame durations increase the impact of protocol overhead and reduce the sleep opportunities. This network comparison model uses a standard 500ms MAC frame duration to reduce data delivery delay and maintain significant network lifetime.

3.7.8 Data Packet Frame Structure

The MSDU data frame structure for the WSN network must reduce the protocol overhead to allow the 128-byte MSDU limit to accommodate the largest possible data payload. The IEEE 802.15.4 standard MAC frame protocol format shown in Figure 3-2 offers a format that can expand from 5 to 25 bytes of protocol overhead depending on the address scheme indicated in the *Frame Control* fields. The data packet MSDU used in the simulation and analytical model employs an 11-byte MSDU protocol to permit data payloads of up to 117 bytes. The 11-byte MSDU header and trailer provides the following fields:

a. Frame Control (2 bytes)

i. Frame Type 3 bits (b0-2)
ii. Security Enabled 1 bit (b3)
iii. Frame Pending 1 bit (b4)
iv. Ack Request 1 bit (b5)
v. Intra-PAN 1 bit (b6)
vi. Reserved 3 bits (b7-9)
vii. Dest. Add Mode 2 bits (b10-11)
viii. Reserved 2 bits (b12-13)
ix. Source Addr Mode 2 bits (b14-15)

b. Sequence Number (1 byte)

c. Address (6 bytes)

i. Source BSS / PAN ID (2 bytes)

ii. Destination BSS / PAN ID (2 bytes)

iii. Source Address (1 byte)

iv. Destination Address (1 byte)

d. FCS Check (2 bytes)

<table>
<tr><th>Octets: 2</th><th>1</th><th>0/2</th><th>0/2/8</th><th>0/2</th><th>0/2/8</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame control</td><td rowspan="2">Sequence number</td><td>Destination PAN identifier</td><td>Destination address</td><td>Source PAN identifier</td><td>Source address</td><td rowspan="2">Frame payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td></tr>
<tr><td colspan="6">MHR</td><td>MAC payload</td><td>MFR</td></tr>
</table>

Figure 3-2 IEEE 802.15.4 MAC Header [WPAN03]

3.7.9 Control Packet Frame Structure

The MSDU control frame structure for the WSN network must limit the protocol overhead to reduce the energy and bandwidth requirements to extend network lifetime. With the exception of the ACK frame, the standard control frame for the WSN simulation and analytical models adds an additional 2-byte duration field to the 11-byte data MSDU header. The IEEE 802.15.4 standard allows a 5-byte PHY-level ACK response which uses temporal context to designate the source and destination addresses.

3.7.10 Interframe Spacing

The IEEE 802.15.4 LR-WPAN sets two levels of interframe spacing to permit the receiver to process and respond to a packet. If a packet is smaller than 18 bytes, the receiver waits a minimum of a short interframe spacing (SIFS) before responding. The SIFS period is set to a minimum of 12 symbols [WPAN03]. Larger packet sizes require a large interframe spacing (LIFS).

The WSN simulation and analytical models extend this two-tiered interframe spacing to include the standard SIFS, PIFS, DIFS, and EIFS explained in Section 2.4.2 to set channel access deference requirements in WLAN and WPAN implementations. The interframe spacings are:

Interframe Spacing	Symbols	Duration
SIFS	12	192 μs
PIFS	16	256 μs
DIFS	20	320 μs
EIFS	54	864 μs

where EIFS represents the maximum allotted time for a receiver to respond with an ACK.

$$(t_{EIFS} = t_{SIFS} + t_{ACK} + t_{DIFS}) \qquad (3\text{-}1)$$

3.7.11 Node Energy Capacity

The latest generation of IEEE 802.15.4 WSN transceiver platforms operates on two AA batteries. Two standard AA-sized lithium batteries can achieve approximately 3000mAh assuming a 2.1 volt cutoff and a 20mA slow drain application [Omn05]. The energy consumption rates for the devices in receive, transmit, and sleep modes are experimentally measured as average current consumption rate in Chapter 7. Given these parameters, the lifetime capacity of a node is set to 3 Amp-hours or 10.8×10^6 mA-seconds.

3.7.12 Energy Consumption Rates

Table 3-1 details the mote platform energy consumption models which were experimentally-obtained in this research by the procedures explained in Chapter 7. Except for

the radio power management (RPM) simulations which use the MICAz parameters, all other simulations use the Tmote parameters. All sleep state energy calculations subtract the low power mode (LPM) transition time from the sleep state duration before calculating the energy cost. The resulting equation is:

$$\text{Sleep\_mA\_seconds} = (t_{SLEEP} - t_{TRANSITION}) * I_{Ave\_Sleep\_Base\_Current} + t_{TRANSITION} * I_{Ave\_Transition\_Current} \quad (3\text{-}2)$$

The node lifetime is determined by the time to consume the 10.8×10^6 mA-seconds battery capacity.

Table 3-1 Tmote and MICAz LPM Transition Responses

Low Power Mode	Total Transition Time (ms)		Average Transition Current (mA)		Average Base Current (mA)		System Effect
	TMote	MICAz	TMote	MICAz	TMote	MICAz	
Receive (RX)	-	-	-	-	21.56	21.97	
Transmit (TX)	-	-	-	-	18.40	19.70	
LPM1: Idle	4.56	4.38	3.72	3.04	0.627	0.743	Freq. Synthesizer Off
LPM2: Power Down	5.15	5.58	2.96	2.94	0.179	0.298	Crystal Oscillator Off Freq. Synthesizer Off
LPM3: Power Off	6.81	5.87	1.88	3.20	0.038	0.190	Voltage Regulator Off Crystal Oscillator Off Freq. Synthesizer Off

3.8 Model Verification and Validation

Does the proposed model realistically represent the actual system? Model verification and validation provide a means to authenticate the relationship between a model and the real system. Verification measures whether the model implements the system assumptions correctly, and validation measures whether the system assumptions adequately model the real system.

3.8.1 Testing Verification

Model verification is the process of determining if a model was implemented correctly. Verification tasks include debugging the OPNET process code and ensuring that the system interfaces work as designed. The GMAC and RPM algorithm simulations were verified by using OPNET's integrated debugging tool and comparing performance metric values with the results generated from the analytical models.

3.8.2 Testing Validation

Model validation ensures the system assumptions used in developing the model are reasonable, and a correctly implemented model would produce results observed in real systems [Jai91]. Three sources provide validity to models: expert intuition, real system measurements, and theoretical results. The GMAC and RPM models were validated using the expert intuition of students and professors knowledgeable in the wireless networking field. Additionally, theoretical results will validate the simulation models using simplified, analytical MATLAB models to provide baseline performance comparisons between the WSN protocols.

3.9 Summary

This chapter has presented the objectives and methodology for developing and evaluating the GMAC protocol, the RPM algorithm, and the symbiotic network. The chapter began in Section 3.1 with an explanation of Jain's ten-step performance evaluation method in order to establish an organized technique for experimental design, measurement, simulation, and modeling. Section 3.2 defined the research problem, goals, assumptions, and boundaries. Sections 3.3 through 3.8 described the evaluation techniques, the simulation scenarios, the performance evaluation metrics, and the model verification and validation techniques used to test the designs. The next chapter provides a detailed description of the GMAC system design.

Chapter 4

GMAC Energy-efficient WSN MAC Protocol

Any intelligent fool can make things bigger and more complex.
It takes a touch of genius
and a lot of courage
to move in the opposite direction.
--Albert Einstein

This chapter presents an energy-efficient sensor MAC protocol developed during this research effort called Gateway MAC (GMAC). The GMAC protocol coordinates transmissions within a cluster and obtains significant energy savings by allowing associated cluster nodes to sleep for extended periods of time. Like the other WSN MAC protocols previously discussed in Chapter 2, all the GMAC nodes have limited resources. The supported sensor application has traffic intended to pass within the cluster to neighboring nodes for intra-network processing and to pass out of the cluster to a network sink for further processing. GMAC has several energy-saving features which not only extend the network lifetime, but also make the network more resistant to denial of sleep attacks.

4.1 GMAC Protocol Overview

Existing WSN MAC protocols extend sensor battery power by placing nodes to sleep during idle periods without network traffic or message exchanges between other nodes. Both of these sleep opportunities require all nodes to remain in the receive mode to monitor all traffic exchanges until an idle timeout occurs. Low-rate IEEE 802.15.4-compliant sensor platforms expend more energy in receive mode than in transmit mode. Nodes gain sleep opportunities after successfully receiving a request to send (RTS) reservation message or timing out after an idle period before the next active cycle. The use of a centralized point coordinator, or access point, in homogeneous wireless sensor networks is generally not employed since the clusters are deployed in an ad hoc manner. GMAC presents a self-configuring, cluster-based MAC protocol which leverages the traffic adaptability of the DCF mode for nodes to reserve time slots and the efficiency of the TDMA schedule-based mode to deliver the messages. With the frame cycle shown in Figure 4-1, nodes with no traffic to send wake up to receive a gateway traffic indication message (GTIM), determine they are not intended receivers, and then return to sleep until the next cycle. Only nodes with queued traffic wake up at the beginning of the contention period. During the contention collection period, transmitting nodes request a scheduled exchange slot from the cluster coordinator, or gateway node, during the subsequent contention-free distribution period. Establishing a collection and distribution traffic rhythm enables nodes to sleep for extended durations and facilitates bi-directional traffic.

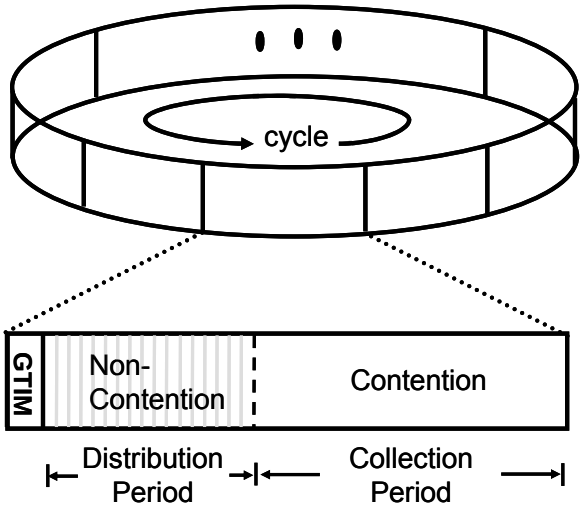

Figure 4-1 GMAC Frame Architecture

Remote sensors are limited in energy; therefore their transmission methods must conserve transmit power by limiting the distance of their transmissions and sleeping while inactive. These two goals of limiting transmission power and maximizing sleep opportunities impede each other because reducing the transmission range for each link requires sensor nodes to cooperate by forwarding more messages for their neighbors. Remaining available as an intermediate routing node to forward messages interferes with opportunities for sleep. One solution is to locally coordinate these functions. GMAC implements a passive cluster coordinator election scheme similar to LEACH [HeC00] presented in Section 2.6.1, but the GTIM algorithm allows for a self-election based on each node's available battery and memory resources, not a strict probability-based calculation. PACT [PeC01], presented in Section 2.6.2, addresses battery resource as a discriminator for cluster head eligibility, but the election is still based on probability, not the available energy level. GMAC's multi-tiered resource levels rotate the cluster coordinator duties among the nodes with the most available resources, and these duty rotations provide graceful network degradation until all nodes' energy levels are exhausted.

The GMAC collection and distribution strategy provides a scalable network for dense sensor fields, promotes fair data exchange, and utilizes the bandwidth efficiently. The dynamic allocation of the contention-free exchange time slots offers the same network scalability as contention-based schemes, but the contention-free period offers better network stability under heavy loads due to the scheduled nature. Since nodes compete equally during the contention period using small future RTS (FRTS) control messages, the pseudo-random exponential backoff promotes fair access. The central gateway node dynamically adjusts the size of the collection and distribution periods according to the inter-network and intra-network traffic patterns to ensure efficient bandwidth utilization.

4.2 GMAC Energy-efficiency Design Strategies

Gateway MAC is a WSN MAC protocol designed to increase network lifetime by maximizing node sleep durations. To attain these benefits, the protocol must address each of the primary sources of WSN energy loss discussed in Section 2.5: idle listening, frame collisions, protocol overhead, and message overhearing.

4.2.1 GMAC Idle Listening Energy-Saving Techniques

While other WSN protocols strive to reduce idle listening, GMAC eliminates network-wide idle listening to obtain significant energy savings. As described in Section 2.5.1, SMAC reduces idle listening time by condensing all traffic into a smaller time period and transitioning to sleep at a designated time for the remainder of the frame. Unfortunately, all nodes must listen to an idle channel after the traffic transactions are completed until the designated sleep time. Idle listening causes SMAC to perform better during heavier traffic conditions when it can take advantage of message passing, NAV sleep opportunities. TMAC, on the other hand, improves on this technique by dynamically setting the start of the sleep time to the completion of network traffic for each frame. When all nodes have sensed the end of traffic for an established timeout period, they automatically transition to sleep until the beginning of the next contention period. Again, this MAC protocol reduces the amount of time all nodes monitor an idle channel down to the timeout period, but it does not eliminate it. GMAC uses an innovative traffic rhythm to eliminate idle listening for all except the gateway node. Using a postal delivery analogy, people with mail to send go out to the curb at a certain time, and a designated postal delivery person collects everyone's mail exchange requests, one letter at a time. The message collection period in Figure 4-2 illustrates this mail collection phase. Those people with no mail to send may stay in bed until mail distribution time. After collecting all of the mail and exchange requests, the postal delivery person forwards all non-local mail out of the neighborhood. At a previously coordinated time, everyone wakes up to hear the postal delivery person announce the schedule for those people who have mail to send, those people who have mail to receive, and when each of those exchanges will take place. Everyone then returns to their house to sleep until the appropriate time to meet for a mail exchange. If a person has no mail to send or to receive, that person may sleep until the next schedule announcement – "mail call." In this scenario, only the postal delivery person will wait at the mailbox for an extra moment to ensure no one else has mail to send. Similarly in GMAC, only the gateway node waits an idle channel timeout period during the message collection period. The GTIM overhead is 14 bytes plus 3 bytes per schedule entry, or less than 10% of the duration of a TMAC adaptive timeout period.

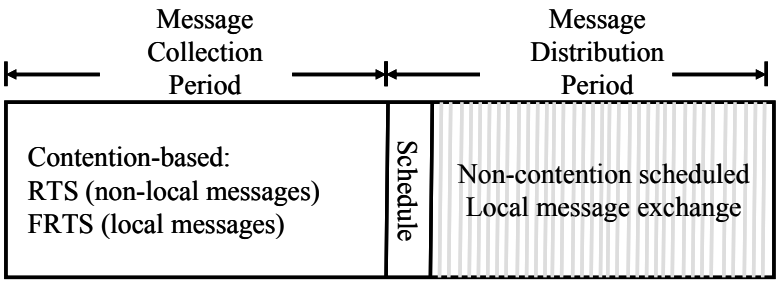

Figure 4-2 GMAC Collection and Distribution

4.2.2 GMAC Frame Collision Energy-Saving Techniques

GMAC improves on frame collision energy losses by utilizing future RTS (FRTS) messages and exchanging the data messages during a non-contention TDMA distribution period. Since nodes sending FRTS messages must contend for the channel, frame collisions will still occur due to nodes simultaneously completing contention backoffs, hidden terminal collisions, and non-zero radio receive-to-transmit switchover times. However, the use of FRTS messages reduces the time and energy costs of the collision by limiting the size of the message collision from a maximum of 128 bytes down to 13 bytes. Once the data exchange schedule is initiated, other nodes cannot penetrate the data exchange and collide with frames since they must wait for either a PIFS or DIFS deference period.

4.2.3 GMAC Protocol Overhead Energy-Saving Techniques

GMAC improves on the protocol overhead by passively electing the gateway node and minimizing the control messages required to produce a traffic-adaptive TDMA schedule. Gateway nodes periodically self-elect themselves in a passive, resource-based contention scheme. The GMAC protocol reduces the amount of control messages by utilizing an algorithm which does not require any knowledge of other nodes' resource levels, yet effectively elects the nodes with the most resources. GMAC's scheduling routine minimizes the complex control overhead of scheduling optimizations like TRAMA discussed in Section 2.5.1 by tasking the gateway node to generate the schedule based upon reported requirements during the collection period. Instead of nodes being allocated slots and generating control messages to request or

release slots, GMAC apportions slots according to the network demands. Additionally, GMAC can optimize the schedule by placing successive node transmissions in adjacent time slots.

4.2.4 GMAC Message Overhearing Energy-Saving Techniques

Finally, GMAC reduces the number of nodes susceptible to message overhearing to only those nodes contending for a transmission request. As described in Section 2.5.4, SMAC and TMAC are only able to mitigate the message overhearing effects for all nodes by transitioning those not involved in the upcoming message transaction to sleep after processing the duration field of RTS or CTS messages. Even if the nodes are able to transition to sleep after receiving the RTS, all nodes must expend the energy to monitor the medium during the contention period and receive each RTS packet. As transceiver data rates continue to increase, the sleep transition time requirements will even prevent protocols from taking advantage of these short-duration, NAV sleep opportunities. GMAC not only mitigates the effects of message overhearing for the nodes involved in message transactions, but it also eliminates message overhearing for all of the other nodes. GMAC nodes contending to pass messages or message exchange requests to the gateway node during the collection period will overhear traffic from the nodes which gain access before them. If the transmitting node is sending inter-network (non-local) traffic for the gateway to forward out of the network, all contending nodes may transition to sleep after receiving the RTS message. However, if the transmitting node is sending an intra-network (local) message exchange request via an FRTS, the entire FRTS-SIFS-ACK exchange is too small to provide sufficient sleep transition time for the other contending nodes. GMAC reduces the costs of the message overhearing by reducing the overheard message size to small control messages and by limiting the effects to only impact transmitting nodes.

The GMAC strategy is to eliminate or reduce the sources of energy loss in wireless sensor networks. Not only does GMAC address each of the primary sources of energy loss in order to extend the WSN network lifetime, but it also obtains significant energy savings by eliminating the network-wide overhearing of contention periods and control messages. Like the mail delivery example, the vigilant work of the gateway node allows the other nodes to remain asleep for longer periods of time.

4.3 GMAC Protocol Design

This section describes the details of the GMAC protocol's major functional components using a top-down design approach.

4.3.1 GMAC Dynamic Collection and Distribution Periods

While other WSN protocols strive to reduce idle listening, GMAC *eliminates* cluster-wide idle listening to obtain significant energy savings. Figure 4-2 illustrates a traffic collection and distribution rhythm which enables nodes to sleep for extended durations, facilitates bi-directional traffic within the cluster, promotes fair data exchange, and utilizes the bandwidth efficiently. The dynamic allocation of the contention-free exchange slots offers the same network scalability as contention-based schemes, but the contention-free period offers better network stability under heavy loads due to the scheduled nature. Since nodes compete equally during the contention period using Future-Request-To-Send (FRTS) control messages, the random exponential backoff promotes fair competition for schedule slots. Starting the frame cycle in the collection period, the cluster coordinator, called the gateway node, collects two types of network traffic requests: intra-network (local) and inter-network (non-local) traffic. Intra-network traffic represents messages exchanged between nodes in the same cluster for sensing coordination or data fusion. The sender transmits a FRTS message to the gateway to reserve a delivery slot in the contention-free distribution period. Inter-network traffic represents messages which originate in the cluster to be forwarded by the gateway to the outside network, messages which originate outside the network to be delivered to a cluster node, or tandem messages traveling through the network. The inter-network sender and gateway exchange an RTS-CTS-data-ACK message sequence for immediate collection. The gateway must limit the amount of inter-network messages it stores due to limited memory capacity. After all transactions are complete, the gateway attempts to forward all traffic out of the cluster and then transitions to sleep. The distribution period begins with all nodes waking up and receiving the gateway traffic indication message (GTIM). In this synchronization message, the gateway declares the current time, the GTIM beacon, and the schedule of message transactions between cluster nodes. The distribution period begins immediately after receiving the GTIM, and the collection period begins immediately upon the completion of the distribution period's scheduled packet exchanges. The GTIM describes the

distribution period traffic exchange slots by source, destination, and relative offset time. If a node is scheduled to transmit or receive a message during the distribution period, the node sleeps until the indicated exchange time, wakes up to exchange the message, and then returns to sleep. If a node is not scheduled to exchange a message, the node transitions to sleep throughout the distribution period. When the contention/collection period begins after the last scheduled exchange, only nodes with traffic to send wake up and request a scheduled exchange slot for the subsequent contention-free distribution period.

The significant energy savings provided by the GMAC traffic pattern are a result of the reduction in the amount of time all nodes must monitor the network. Receiving the GTIM is the only time that all nodes will be awake unless the GTIM schedule contains a broadcast message. Unlike the other WSN MAC protocols, GMAC eliminates cluster-wide idle listening and extends the length of time inactive nodes can sleep.

Figure 4-3 illustrates the intra-network traffic exchanges between sender 1 and receiver 3 and the inter-network traffic sent from sender 2 to the gateway. After the start of the collection period, sender 1 calculated a lower exponential contention backoff number and gained access to the medium after a short wait. Upon completion of sender 1's FRTS-ACK exchange with the gateway, sender 2 continues contending for the channel by restarting its backoff countdown timer. Once completed, sender 2 accesses the channel and transmits the inter-network data to the gateway node using the RTS-CTS-data-ACK sequence. After waiting for an idle timeout (TA) period, the gateway node attempts to forward sender 2's inter-network data out of the network during the inter-network traffic and sleep (I&S) period. Note that the intended receiver, receiver 3, and all other nodes which did not have traffic to send were able to sleep throughout the entire collection period.

4.3.2 RAVE: Resource Adaptive Voluntary Election

GMAC periodically elects a new gateway node to equally distribute the energy requirements among all of the nodes using the resource adaptive voluntary election (RAVE) scheme. RAVE is a passive cluster coordinator election scheme similar to LEACH [HeC00], but

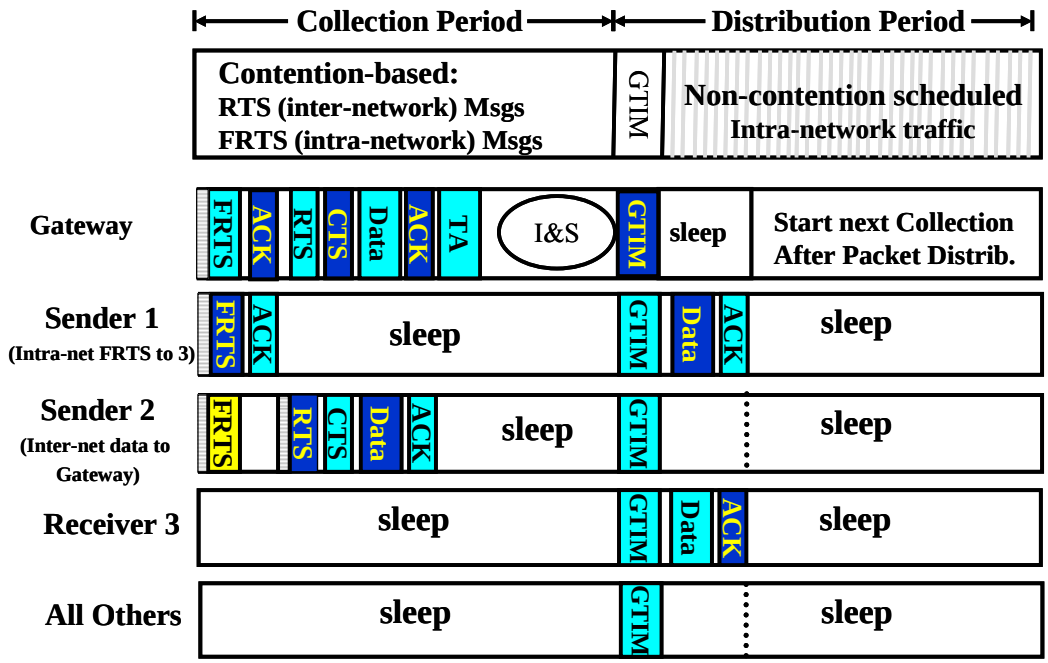

Figure 4-3 GMAC Message Exchange

the RAVE algorithm allows for a self-election based on each node's available battery and memory resources, not a strict probability-based calculation. PACT [PeC01], another passive election scheme, addresses battery resource as a discriminator for cluster head eligibility, but the election is still based on probability.

Although the network has a default changeover frequency, every gateway traffic indication message (GTIM) contains an election bit that is set to indicate whether the network will immediately elect a new gateway. To reduce the overhead of exchanging available resource updates, GMAC uses a passive method of determining the next, most eligible, gateway by calculating an election contention backoff period based upon a node's available resources. Nodes with fewer resources will have longer backoff times. Immediately after processing the election GTIM message, Figure 4-4 shows how each node counts down its self-generated election contention backoff value. One of the nodes in the group which has the most available resources will become the next gateway. The new gateway is the *volunteer* node which first gains contention of the medium after the start of the GTIM period. A gateway node will signal for a new election whenever it transitions to a predetermined lower energy threshold, reaches critically low memory levels, or approaches a default changeover time. Following the election

confirmation sent by the outgoing gateway, all nodes process the GTIM relative schedule times embedded in the previous election GTIM.

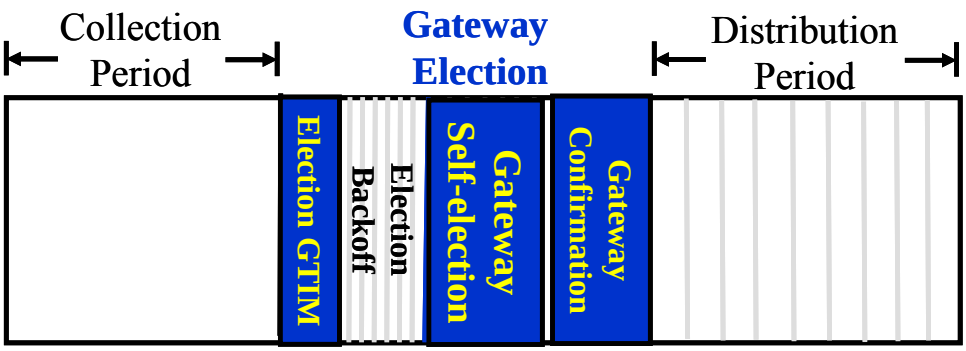

Figure 4-4 GTIM Passive Election Backoff Scheme

GMAC's multi-tiered resource levels shown in Tables 4-1 and 4-2 facilitate the rotation of the gateway duties among the nodes with the most available resources. These duty rotations provide graceful network degradation until all nodes' energy levels are exhausted. The critical resource level algorithm assigns an individual node's resource level (RL) according to the most critical resource. Although this model only shows four distinct resource levels, the model can be extended for better resource resolution.

Table 4-1 Battery Resource Level

Battery Power Level	Power Level Nomenclature	Voltage Range (volts)
00	High	$2.6 < Pwr \leq (3.0\text{-}3.6)$
01	Med	$2.4 < Pwr \leq 2.6$
10	Low	$2.1 < Pwr \leq 2.4$
11	Min	$Pwr \leq 2.1$

Table 4-2 Memory Resource Level

Memory Capacity Level	Memory Level Nomenclature	Percentile Available Memory Capacity
00	High	$30\% < Mem$
01	Med	$20\% < Mem \leq 30\%$
10	Low	$10\% < Mem \leq 20\%$
11	Min	$Mem \leq 10\%$

Critical Resource Level Algorithm

if Pwr=Min or Memory=Min then Resource Level = 3 (one resource is low)
elseif Pwr=Low or Memory=Low then Resource Level = 2
elseif Pwr=Med or Memory=Med then Resource Level = 1
else Resource Level = 0 (both resources are high)

Again, GMAC uses the GTIM election flag bit to indicate the initiation of an immediate gateway election. To reduce the overhead of exchanging available resource updates, GMAC uses a passive method of determining the next gateway by calculating an election contention backoff period based upon a node's available resources. RAVE's election contention backoff algorithm chooses a gateway from the most energy or memory eligible group of nodes using the equation:

$$\text{ElectionBackoff} = \text{Random}\ (2^7) + (\text{RL} * 128) \qquad (4\text{-}1)$$

where ***ElectionBackoff*** is the number of contention slots a node will backoff before sending a self-election packet, ***Random (2^7)*** generates a random number from 0 to 127, and ***RL*128*** offsets the random number into an eligibility band based upon available resource levels (RL). Table 4-3 illustrates the election eligibility contention backoff windows and eligibility groups.

A gateway node will signal for a new election whenever it transitions to a lower energy state, reaches critical memory levels, or approaches a default changeover. Nodes immediately calculate an election contention backoff when they encounter a signaled election. The new gateway is the *volunteer* node which successfully transmits a self-election message after the start of the GTIM period. The departing gateway node confirms the new gateway and changes to regular node status. In the event of a gateway node failure, after waiting for three consecutive missed GTIMs, the nodes will automatically conduct an election with a peer-confirmation mechanism. RAVE also uses this timeout driven peer-election method to initially self-configure the cluster.

Table 4-3 RAVE Election Contention Backoff

Resource Level (RL)	Election Contention Backoff Random (2^7) + (RL * 128)
0 High	0 to 127 slots (or 0ms to 2ms)
1 Med	128 to 255 slots (or 2ms to 4ms)
2 Low	256 to 383 slots (or 4ms to 6ms)
3 Min	384 to 512 slots (or 6ms to 8ms)

4.3.3 Gateway Intra-network Traffic Scheduling

GMAC achieves significant energy savings in both heavy- and light-density traffic environments by performing all required traffic scheduling operations while most of the nodes are sleeping. As discussed earlier in Section 4.3.1, nodes request intra-network schedule message exchange slots by sending the gateway node an FRTS during the collection phase. The gateway node collects all of the requests and generates a schedule based upon a first-come-first-serve (FCFS) priority scheme. Additional options may include creating a message priority QOS system and optimizing the schedule by placing multiple receivers or transmitters into adjacent time slots. The gateway can elegantly optimize the schedule with its limited processing capabilities by first sorting the final schedule list by the destination addresses, and then sorting again based upon source addresses.

Each entry in the gateway traffic indication map (GTIM) lists the source address (1 byte), the destination address (1 byte), and the schedule offset duration (1 byte). The duration of each data packet is quantized in 32µs slots to reduce the size of the offset duration field. The slot size is relatively small to ensure that the maximum residual quantization error plus the SIFS spacing would not equal or exceed a PIFS time to maintain the integrity of the schedule. PIFS-sized schedule guard band slots would allow another gateway node to interrupt the schedule. Each node wakes up a SIFS time before the scheduled transaction. In the event that a scheduled exchange is not acknowledged for any reason, the schedule will continue. The unacknowledged sender will submit a new FRTS reservation in the following collection period. The design

decision to prohibit immediate schedule retransmissions is based upon the need for protocol simplicity with limited WSN processing and program storage capabilities.

Once the nodes receive the schedule, each node uses the parsing scheme detailed in Appendix D to accumulate sleep until its next exchange event. This parsing scheme uses a minimal amount of state information to incrementally parse a schedule of any size and sequentially employ system interrupts to synchronize maximum-sized sleep opportunities.

4.3.4 Gateway Traffic Indication Message (GTIM) Beacon

After collecting the intra-network traffic schedule requests, the gateway node assembles the scheduling along with other network management information into the gateway traffic information message (GTIM). The GTIM includes the standard IEEE 802.15.4 PHY (6 bytes) and MSDU shown in Figure 4.5.

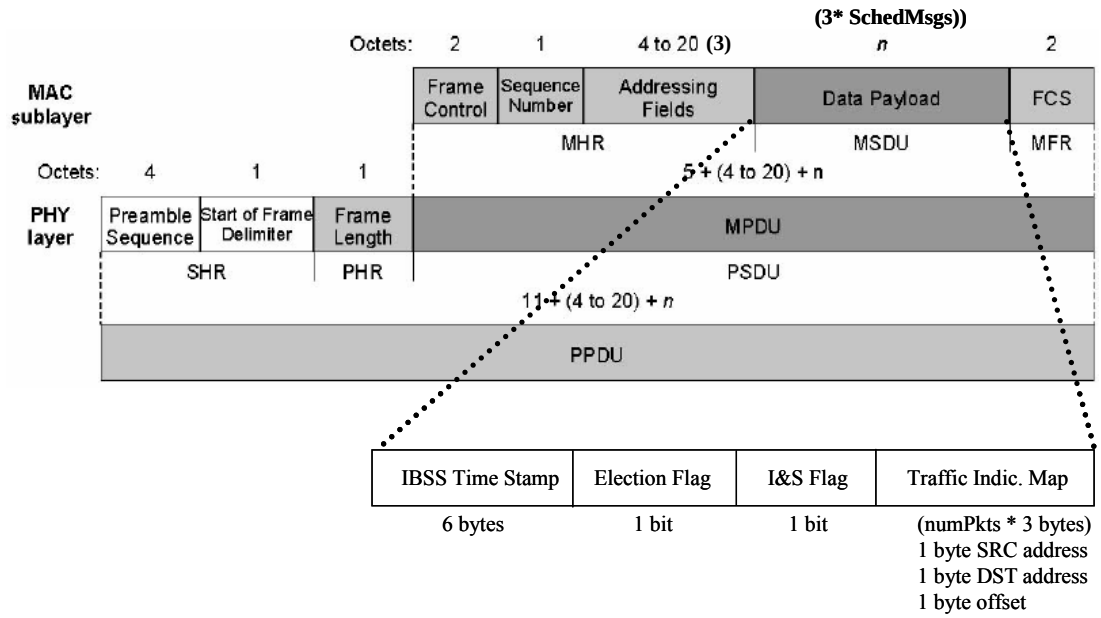

Figure 4-5 GTIM PHY/MAC Frame Header [WPAN03]

The final size of the GTIM data payload depends upon the number of scheduled messages. As described in the previous section, each schedule entry takes three bytes to list the source, destination, and schedule offset start time. The bottom data payload frame in Figure 4-5 depicts

the GTIM fields. The dynamic GTIM message length is:

2 byte frame control + 1 byte sequence + 3 byte address (2 BSS, 1 dst) + 6 byte absolute time + 2 byte FCS + (n*3 byte) GTIM payload
= 14 bytes + 3 * numScheduledMessages

$$GTIM\ Message\ Length = 14\,bytes + (3\,bytes * numScheduledMessages) \qquad (4\text{-}2)$$

4.3.5 Link Layer Security

Wireless sensor networks offer the ability for applications to remotely monitor and react to events, but their remoteness also introduces challenges and vulnerabilities for network control and energy consumption. The most significant WSN MAC layer energy vulnerability is a form of denial of service attack called the denial of sleep attack. A denial of sleep attack penetrates a device's power management system to reduce the opportunities to transition into lower power states. The security offered by the current software and hardware implementations is insufficient to protect a WSN from a denial of sleep attack. The three basic WSN security options are to encrypt the data and the header, encrypt the data, or provide no encryption. The main goal of a denial of sleep attack is to force a sensor node to stay awake and receive a counterfeit transmitted packet. If the complete packet is encrypted, the sensor node must receive the entire packet, decrypt the header, and then determine that it is not a receiver. The data-only encryption mode allows the node to view the header as it arrives, but the node will not be able to authenticate the sender until the packet data is decrypted. In this case, if the attacker is able to provide a legitimate source and destination, the receiver will stay awake to accept the entire packet. The link layer will discard any packets which fail the message authentication code check. The unencrypted mode expends the same amount of energy receiving the packet, but it will accept the incoming message and pass it up to the network layer.

Achieving a secure system requires security integration into every component to prevent a vulnerable point of attack [PeS04][PoH04]. WSN designers must incorporate protecting the critical energy resource into the system architectures. GMAC incorporates the CC2420 transceiver's hardware AES security operations [Chi04] into a centralized architecture to protect the network against denial of service attacks. As discussed in Section 2.7.2, the CC2420 offers four security modes:

- Disabled

- Cipher Block Chaining (CBC-MAC) authentication
- Counter (CTR) encryption / decryption
- Counter with Cipher Block Chaining-Message Authentication Code (CCM) authentication and encryption / decryption.

GMAC uses the CBC-MAC authentication to screen all messages requesting access to the network. The gateway node acts as a network sentry to shield the sleeping nodes from messages designed to drain their energy resources. Since cluster nodes only respond to the gateway node, network attackers cannot penetrate the link layer of the GMAC protocol. Unicast or broadcast messages sent to the gateway must be authenticated prior to being distributed to the individual nodes. The most success that a broadcast attacker can achieve is by sending broadcast messages to the gateway, forcing the gateway to receive the entire message before discarding it due to authentication failure. Once the gateway's energy level reduces to the next lower critical resource increment, a fresh gateway takes its place. The attacker must effectively erode the network's energy one node at a time.

4.3.6 Protocol Timing Considerations

TDMA protocols normally require strict timing synchronization to effectively coordinate communications and to efficiently utilize the available bandwidth. The low-rate WSN deployment experiments have all used contention-based protocols for simplicity, reliability, and adaptability. In order for the GMAC protocol to operate in remote deployments, the TDMA synchronization requirements must be similar to the contention-based schemes. The following sections present GMAC's synchronization design considerations.

4.3.6.1 GTIM Beacon Time

The GMAC protocol requires relative, not absolute, time synchronization in order to coordinate the network activities. With a 6-byte, approximate nine year revolving clock, the protocol retains the ability to maintain absolute time if additional mechanisms, like global positioning system (GPS), are added to the network to prevent post-deployment time drift. GMAC synchronizes each node to the network time through the 6-byte time field in the GTIM sent by the gateway at the beginning of every distribution period. Without accurate feedback from a reliable clock source, the network time will continuously drift with the gateway node's

clock. With the gateway broadcasting a network time stamp every 500ms in the GTIM, the inexpensive WSN platform clocks are able to adequately synchronize the network with the relative time.

Every GMAC node must be able to stay within a SIFS period in order to properly coordinate the network. The longest opportunity to drift is the time from the completion of the GTIM to the start of the next GTIM for nodes with no traffic to send or receive. Using the experimental data from the TSPN protocol presented in Section 2.8 [GaK03], the synchronization period required to maintain a 200µs SIFS period accuracy is:

$$T_{synchPeriod} = \frac{T_{accuracy} - T_{synchError}}{T_{drift}} = \frac{200\mu s - 50\mu s}{4.75\mu s} = 31.6 \text{ sec} \tag{4-2}$$

The specified clock drift, T_{drift}, and the worst case synchronization error for a given protocol, $T_{syncError}$, are used to determine the required synchronization period ($T_{synchPeriod}$) to achieve a target synchronization accuracy bound ($T_{accuracy}$). The result shown in equation 4-2 allows the GMAC frame period to extend to more than 30 seconds and still resynchronize back to the worst case synchronization error of 50µs just prior to the start of the contention-free distribution period. The supported wireless sensor network application program may also have latency requirements which would affect the maximum size of the frame period.

4.3.6.2 Schedule Guard Band

The time between successive message transmissions during the distribution period must be sufficient to prevent overlapping transmissions, but not allow the loss of control of the channel. The duration of each data packet is quantized in 32µs slots to reduce the size of the offset duration field. The 1-byte duration slot size is relatively small to ensure that the maximum residual quantization error plus a SIFS spacing would not equal or exceed a PIFS time to maintain the integrity of the schedule. PIFS-sized schedule guard band slots would allow another gateway node to interrupt the schedule. The SIFS early wake up and guard band monitoring eases the timing constraints to synchronize the TDMA message exchanges.

4.3.7 Multi-hop Considerations

The purpose of WSN networks is to collect data in a remote location for applications which monitor and react to events. Although this research studied a single-hop, WSN network to optimize the energy efficiency of the MAC protocol at the data link layer, GMAC has been designed in a modular fashion to seamlessly integrate into the standard protocol stack. This section describes how GMAC adapts to a multi-cluster network.

4.3.7.1 Multiple-hop Network Routing

Although most of the design efforts focus on the data link-layer issues, GMAC also supports network routing in a multiple-hop network. Figure 4-6 illustrates how GMAC integrates into a two-tiered system to route network traffic in a wireless distribution system. Many ad hoc network routing protocols, such as the ad hoc on-demand distance vector (AODV) routing protocol, can be combined with a GMAC module to pass traffic across a multi-hop network. The gateway node accepts and relays traffic based upon the 2-byte destination BSS address. Any network layer module can provide the network routing table for GMAC to access in order to determine the packet's next hop.

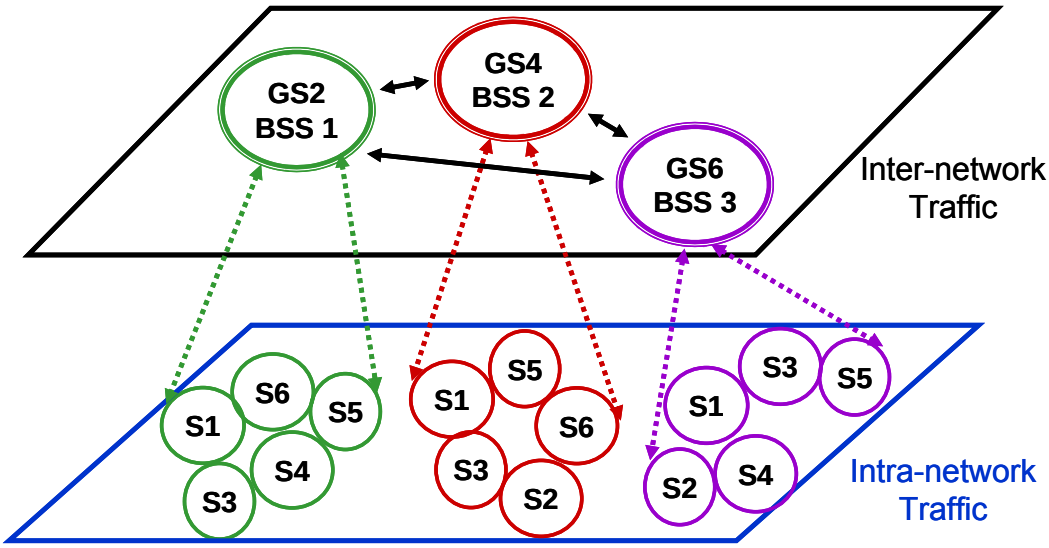

Figure 4-6 Two-tiered Network Routing

4.3.7.2 Multiple-hop Scheduling Cells

Adjacent GMAC clusters will automatically disperse their scheduled distribution times based upon the current traffic conditions. Figure 4-7 illustrates a multiple-BSS scenario which assumes that two individual clusters, BSS1 and BSS2, begin with approximately the same GTIM beacon time. Nodes in both clusters compete with one another during the first contention period to submit schedule reservations. Multiple requests will successfully intermingle because the collection period uses a standard DCF access policy. Sender BSS1 gains access to the medium first and sends a reservation request to the gateway BSS1. Both of the active nodes in BSS2 overhear the BSS1 traffic and continue to wait to gain channel access. Next, the sender BSS2 gains access and sends a reservation request to gateway BSS2. Finally, each gateway waits a TA timeout to ensure all intra- and inter-network transactions are complete and transitions to sleep. The inter-network and sleep (I&S) period provides an opportunity for gateways to exchange inter-networking traffic or sleep to conserve energy.

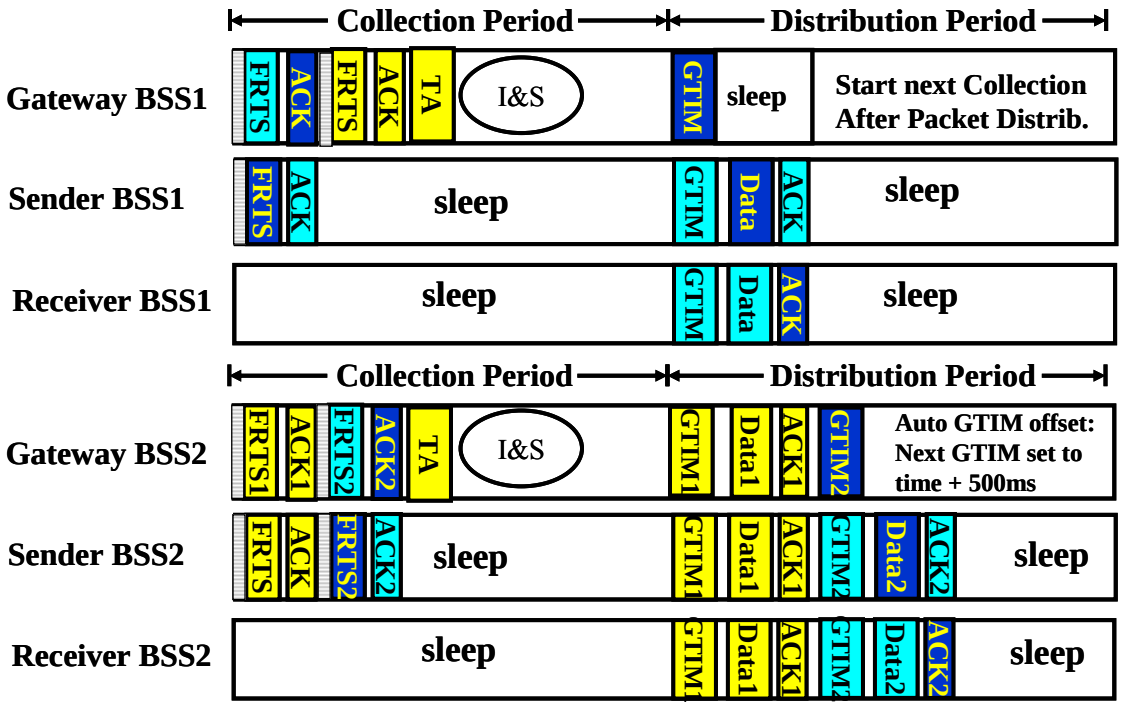

Figure 4-7 Multi-Cluster Network Auto-GTIM Offset

All regular nodes in both networks wake up one SIFS interval before their scheduled BSS GTIM. Assuming the BSS1 GTIM time is scheduled slightly before the BSS2 GTIM time, the BSS2

gateway and associated WSN nodes will overhear the BSS1 GTIM and the entire schedule for the first iteration. Once the entire BSS1 schedule is complete, the BSS2 gateway will transmit its GTIM and initiate the BSS2 schedule. Because all GTIM times are relative, the next BSS2 GTIM time will occur 500ms after the last successful GTIM transmission and now be automatically staggered to occur after the BSS1 distribution period. As the BSS1 schedule expands and contracts, the two schedules will dynamically optimize their separation based upon the BSS traffic requirements.

4.3.7.3 Multiple-hop Inter-network Traffic Delivery

Since the primary purpose of sensor networks is to collect, process, and deliver data, WSN networks must be able to provide a wireless distribution system to forward this data to the appropriate application. Figure 4-8 shows how a BSS1 gateway node collects this inter-network traffic from an associated node and forwards the data to the gateway node in BSS2. Chapter 6 will show that a gateway node managing a 50 node cluster can remain awake throughout the entire period and still maintain a longer network lifetime than TMAC and SMAC. Allowing the gateway to sleep after an adaptive timeout period for monitoring inter-network traffic significantly adds to the network lifetime, but it adds to the latency of forwarding multiple-hop traffic. The network designer must balance the application requirement for multi-hop latency for network lifetime.

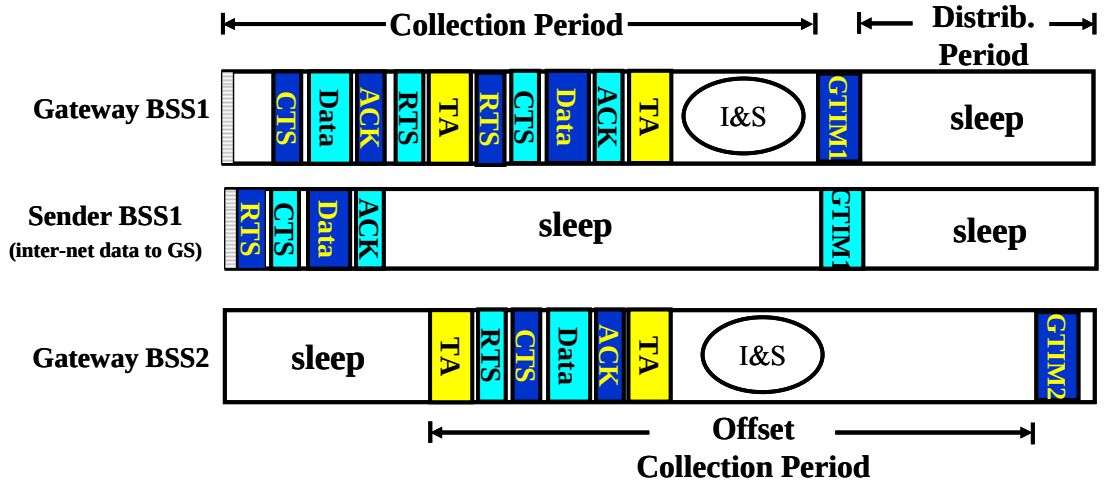

Figure 4-8 Inter-Network Message Exchange

4.4 WSN Protocol Qualitative Network Comparisons

GMAC's elimination of network-wide idle listening produces significant energy savings. The figures in this section illustrate each protocol's energy-saving techniques in order to compare their strengths and weaknesses.

4.4.1 Empty Network Comparisons

Figure 4-9 illustrates the capability of each protocol to save energy through powering down the radio during sleep intervals in an empty or ultra-low network traffic scenario. SMAC employs a static listening period to concentrate all transmissions which would normally occur throughout a frame into a small, typically 10%, duty cycle. Although SMAC reduces idle listening costs by as much as nine times the IEEE 802.11 (active) case, the fixed duty cycle does not adapt to changing traffic patterns to achieve even more savings or utilize bandwidth more efficiently. TMAC builds on SMAC's success by changing the static active listening period into a dynamic active period and transitioning all nodes to sleep after the network traffic is complete. As was discussed in Section 2.5.1.2, the network adaptive timeout (TA) is determined by the longest possible time it would take a hidden terminal node to sense traffic on the channel ($1.5*[t_{CWMAX} + t_{RTS} + t_{SIFS}]$) . This adaptive timeout is very effective at reducing idle listening, but every node in the network must actively monitor the channel throughout this idle period. With current radio parameters, the timeout period is modeled as 13.48 ms. With a 500ms frame time, this timeout produces a 2.7% duty cycle for an empty or ultra-low traffic scenario. The BMAC protocol requires every node to periodically wake up and sample the network using low power listening. With the capability to take a reliable sample in 350ms, BMAC is able to achieve as low as a 1.5% duty cycle in small networks. This active duty cycle rises to 2% for 20 node networks and 3% for 60 node networks [PoH04]. As described in the previous sections, GMAC achieves network energy savings by allowing all inactive nodes to sleep for the longest duration possible. Only the gateway node or sensor (GS) and transmitting nodes are active during the collection time. In the empty network case, the gateway node uses the same timeout period as the TMAC nodes in order to sense network traffic during both the collection and distribution periods while all other nodes sleep. With current technology, the 14-byte empty

GTIM message and SIFS early wakeup takes 832 µs and produces a 0.2% duty cycle in a 500ms frame. GMAC is thus able to eliminate network-wide idle listening at the cost of the GTIM broadcasting the network schedule and synchronization message.

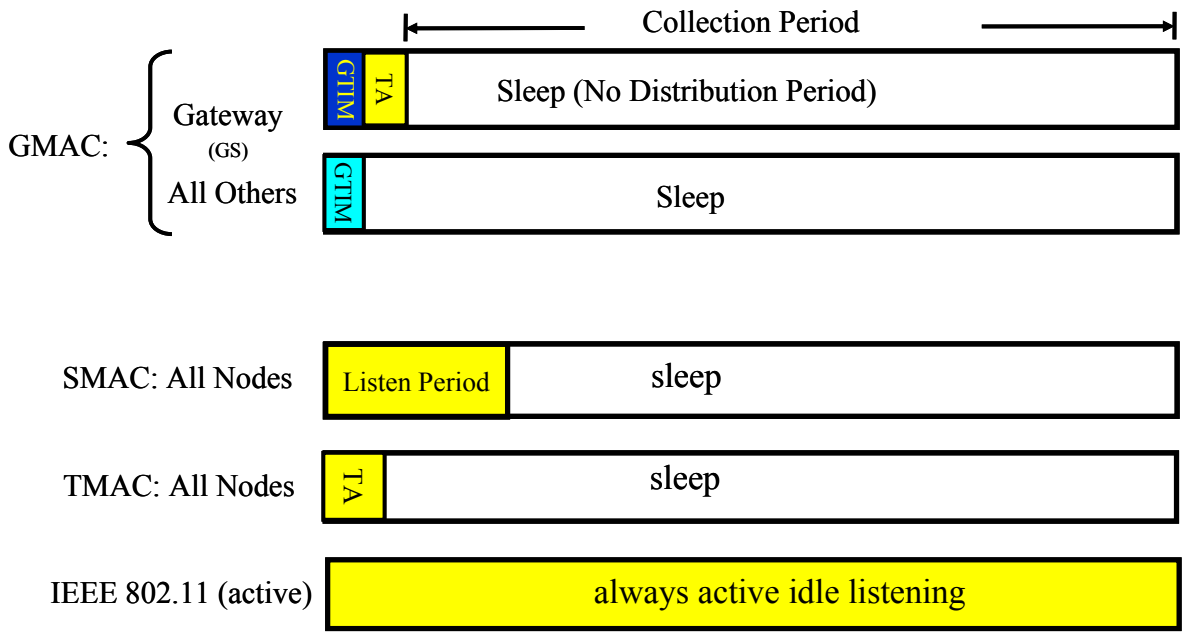

Figure 4-9 WSN Empty Network Protocol Comparisons

The IEEE 802.11 DCF in power save (PS) mode described in Section 2.4.2 has the ability to save the associated nodes a considerable amount of energy by storing the BSS messages and allowing the nodes to sleep through multiple beacon periods. The IEEE 802.11 PS protocol could be adapted to allow the AP to sleep and conserve energy after conducting the network transactions, but the AP would still require additional memory assets to provide message storage. Also, the AP would require additional energy resources because the management responsibilities are not rotated among network nodes. Figure 4-10 illustrates how the AP sends a periodic beacon like GMAC, and the associated nodes can gain even more energy savings by periodically waking up after multiple beacons periods.

The IEEE 802.11 ad hoc protocol displays its energy inefficiency in the empty network scenario. Figure 4-10 also shows how every node must wake up, receive a TBTT beacon, and wait for an entire ATIM window period before transitioning to sleep. Like SMAC, this ATIM window period is statically set upon network deployment based upon anticipated network traffic conditions.

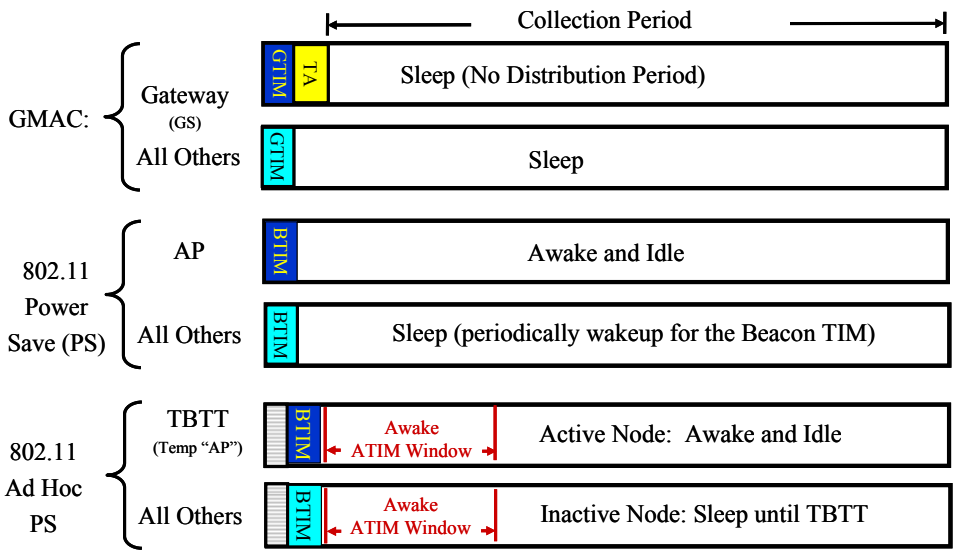

Figure 4-10 802.11 and GMAC Empty Network Lifetime in WSN Ad Hoc Environment

4.4.2 Unicast Traffic Network Comparisons

Figures 4-11 through 4-17 illustrate the capability of each protocol to save energy by powering down the radio after conducting unicast message traffic. Figure 4-11 provides the IEEE 802.11 active model for comparison. All nodes constantly monitor the channel.

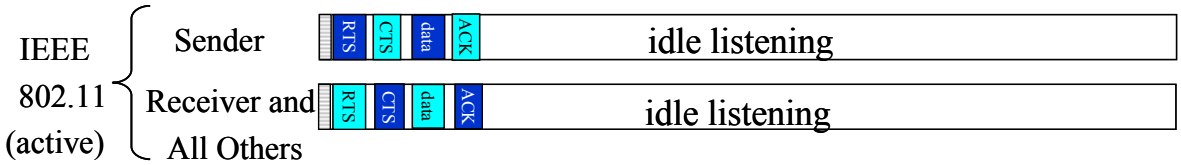

Figure 4-11 IEEE 802.11 Unicast Message Network Transaction

The SMAC protocol employs a static listening period to concentrate all transmissions which would normally occur throughout a frame into a small duty cycle. Figure 4-12 illustrates

a typical RTS-CTS-data-ACK unicast transaction. The ability for all *other* nodes to avoid message overhearing by transitioning to sleep after receiving the RTS duration field enables SMAC to consume less network energy during active message traffic than in the empty traffic network. The fixed active period continues to limit SMAC's energy-saving capabilities.

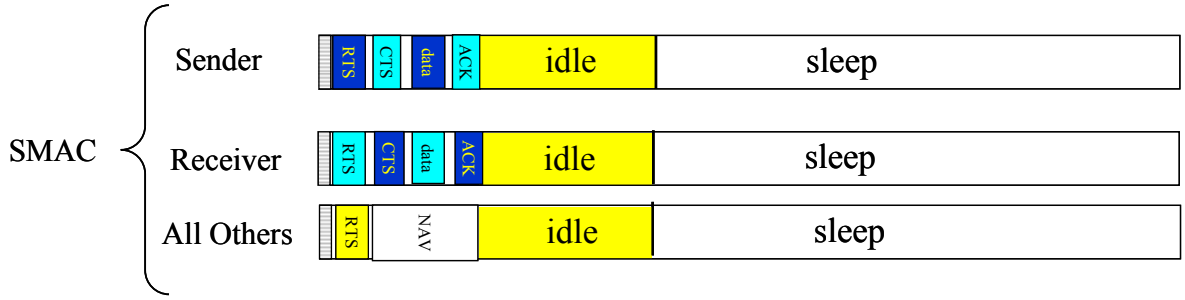

Figure 4-12 SMAC Unicast Message Network Transaction

The TMAC protocol's adaptive timeout (TA) dynamically adjusts the sleep period according to the traffic requirements. Shown in Figure 4-13, TMAC also avoids message overhearing by allowing nodes to transition to sleep during other node traffic exchanges. Unlike GMAC, all TMAC nodes must monitor the channel during contention periods and RTS messages. Additionally, requiring all nodes to monitor the idle channel for the entire timeout period (TA) consumes a significant amount of energy.

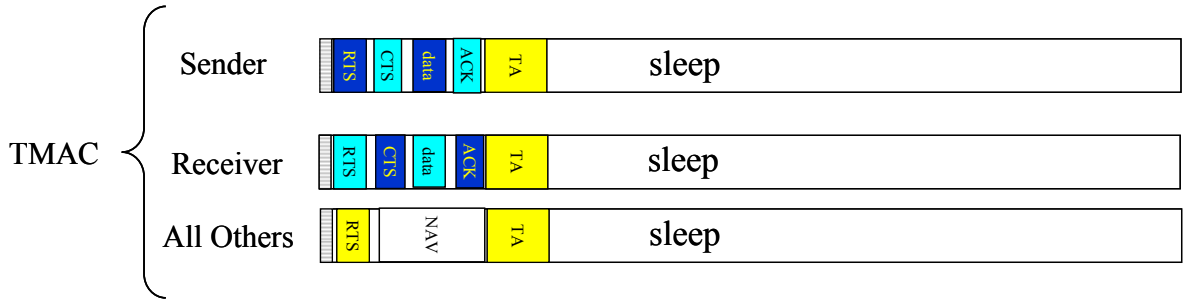

Figure 4-13 TMAC Unicast Message Network Transaction

The BMAC protocol relies on ultra-low traffic patterns to offset the energy costs for all nodes to receive extended preambles and every message. As shown in Figure 4-14, every node will average receiving one-half of the extended, full-frame preamble. Every node must also receive the entire packet before determining the intended receiver. In networks with regular

traffic, BMAC expends significantly more energy than the other energy-efficient WSN MAC protocol models due to message overhearing effects.

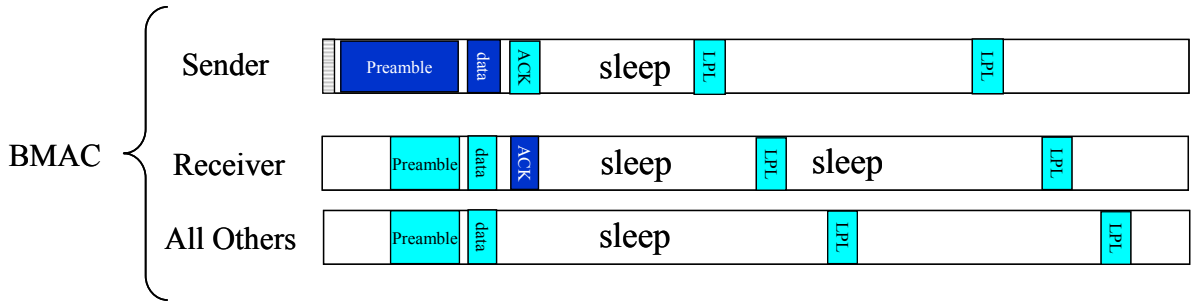

Figure 4-14 BMAC Unicast Message Network Transaction

The IEEE 802.11 DCF PS unicast transactions shown below in Figure 4-15 illustrate the minimal additional cost a node returning from power save mode must expend to request queued data. A small PS Poll control packet is sent to the AP if a node detects queued data in the beacon traffic indication map (TIM). A node will continue the PS Poll – data –ACK sequence until the AP resets the data frame control *MORE DATA* field to zero. The actual IEEE 802.11 protocol specification can require a large beacon TIM to account for an individual bit field for every associated node. Allowing the TIM to be reasonably-sized and not considering the additional memory sizes, the IEEE 802.11DCF can effectively save energy in the WSN network environment.

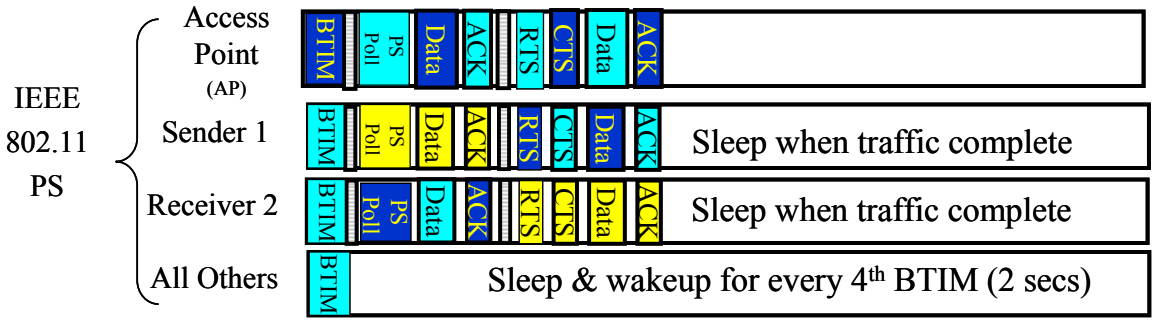

Figure 4-15 IEEE 802.11 DCF PS Unicast Message Network Transaction

The IEEE 802.11 ad hoc power save mode transaction shown below in Figure 4-16 illustrates the inefficiency of the target beacon transmission time (TBTT) contention period and the ATIM window. Like SMAC, the protocol overhead of the fixed ATIM window consumes

too much energy to compete with the other WSN protocols. Allowing the ad hoc protocol to set an adaptive ATIM window timeout similar to TMAC would improve the energy efficiency, but the protocol still implements both a beacon and network-wide adaptive timeout, combining both the protocol overhead effects of GMAC and TMAC. Additionally, the ad hoc PS protocol requires all *active* nodes to remain awake throughout the entire frame cycle. In the example illustrated in Figure 4-16, the TBTT temporary AP, the sender, and the receiver would remain awake throughout the entire cycle. For fair WSN comparison, the sender and receiver should immediately transition to sleep, and the TBTT temporary AP should be allowed to transition to sleep after an adaptive timeout. Even provided all of the sleep transition advantages, the network-wide idle listening for the contention backoff of the TBTT, the beacon, and the ATIM window make the IEEE 802.11 ad hoc PS mode unable to compete with the other WSN energy-efficient protocols. In fairness to the intended network protocol environment, this IEEE 802.11 ad hoc PS network protocol would operate better than the quasi-static WSN protocols in a mobile ad hoc network where the clusters are more decentralized. Even the decentralized TMAC protocol requires virtual clusters in order to synchronize their wakeup times.

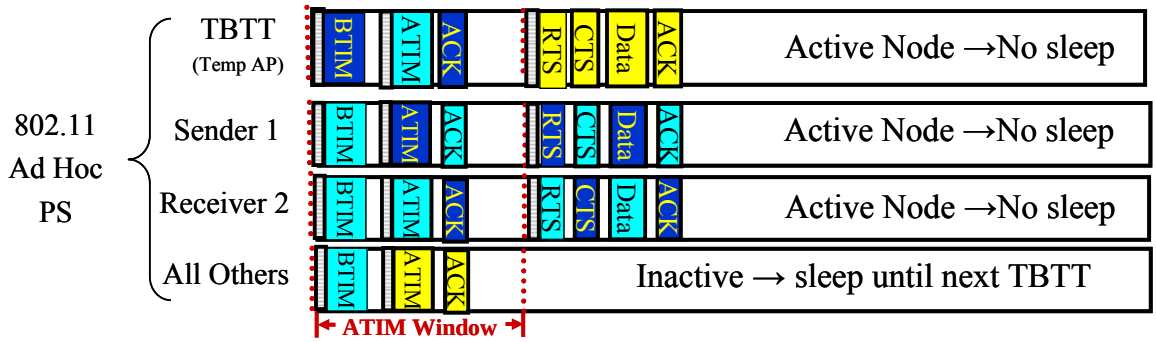

Figure 4-16 IEEE 802.11 Ad Hoc PS Unicast Message Network Transaction

The GMAC protocol's energy-efficient techniques produce significant network energy savings in unicast message scenarios by only requiring the nodes involved in message exchanges to be active during the collection and distribution periods. All other nodes wake up only to receive the GTIM's schedule and synchronization beacon. Figure 4-17 shows that a sending node wakes up during the collection period, contends for the medium, and requests a message exchange time slot for the distribution period. Unlike many of the other protocols, only a small number of nodes participate in the energy-consuming channel contention period during the

collection phase – the gateway node and the sending nodes. Once sending nodes have successfully requested a message exchange time slot, they transition to sleep. All nodes wake to receive the GTIM. Throughout the remainder of the distribution period, only the sending and receiving nodes are awake during the message exchanges. The significant energy savings of GMAC are illustrated in the bottom row of Figure 4-17. Any node not participating in a message exchange sleeps for the complete frame except to receive the GTIM. GMAC effectively creates a traffic rhythm which allows most nodes to sleep for an extended duration of time.

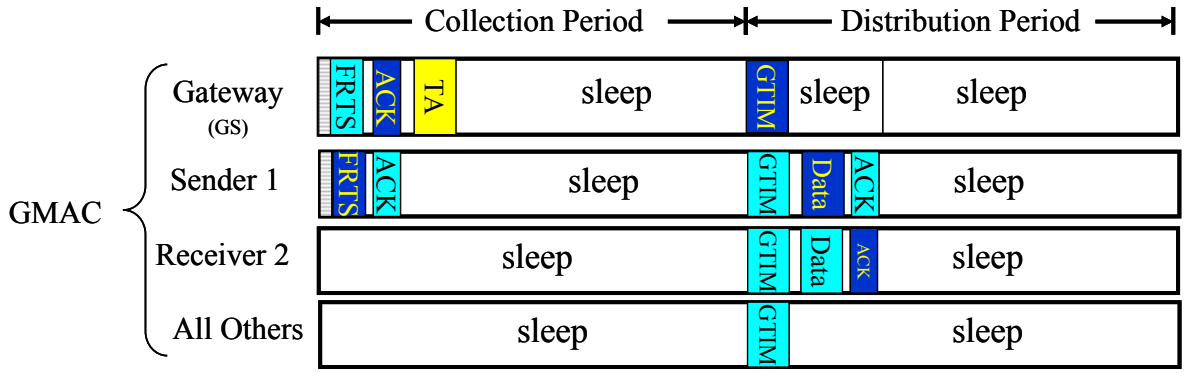

Figure 4-17 GMAC Unicast Message Network Transaction

GMAC also provides energy savings for broadcast traffic scenarios; however, since every node must be awake to receive the broadcast message, these savings are not as significant as the unicast traffic scenarios.

4.5 Summary

This chapter has presented the system design for the GMAC protocol. The chapter began in Section 4.1 by presenting a GMAC protocol overview. Section 4.2 further described how this protocol addressed each of the primary sources of wireless sensor network MAC protocol energy loss. Section 4.3 followed with a thorough functional description of each major GMAC protocol design component. Then, all of the energy-efficient WSN MAC protocols were qualitatively compared in Section 4.4 to illustrate GMAC's significant energy-saving advantages. The next chapter describes the analytical and simulation models developed to compare the performance of each of the WSN and IEEE 802.11 protocols.

Chapter 5

Analytical and Simulation Models

Part of the inhumanity of the computer is that,

Once it is properly programmed and working smoothly,

it is completely honest.

--Isaac Azimov

This chapter presents the analytical and simulation models used to compare the WSN and WLAN protocols with the network performance metrics. Complete MAC and PHY layer models were developed in OPNET Modeler 11.0 for the SMAC, TMAC, and GMAC protocols. An analytical model for each was also created in MATLAB to verify the simulation and extend some of the comparison tests.

5.1 MAC Analytical Modeling Performance Analysis

Analytical models provide an inexpensive and rapid means to evaluate the performance of a proposed system and conduct initial parameter tradeoff analysis. Analytical models lack sufficient detail to provide complete system comparisons; but, if the simplifying assumptions are

fair, the results provide insight into the scale of system comparisons. This section details the analytical MAC protocol models for WLAN IEEE 802.11, SMAC, TMAC, and GMAC under unicast and broadcast traffic conditions. The specific equations for each of the modeled protocols are listed in Appendix E.

5.1.1 General Analytical Model

The analytical models implemented in MATLAB required assumptions to reduce the complexity of the system under study. The primary strategy utilized in the WSN analytical network models was to classify each node into one of four categories: transmitter, receiver, gateway/AP, and idle nodes. Having one transmitter node represent all network transmitting nodes makes network traffic scaling tractable, and the only limiting factor is the inability to model frame collisions. The effects of message overhearing in GMAC and the IEEE 802.11 AP PS mode are taken into account to model the decreasing number of transmitting nodes waiting to gain contention as each transmission during the collection period is serviced. The following summation equation accounts for these decreasing message overhearing nodes:

$$Number\ of\ Overheard\ Transmissions = \sum_{n=1}^{n=num\_Pkts-1} n = \frac{num\_Pkts*(num\_Pkts-1)}{2} \tag{5-1}$$

Each initial transmission requires a medium contention time modeled by a DIFS and the average minimum contention window ($CW_{min}/2$ * slot time = 15 * slot time).

A similar method was employed to have one node represent all of the receiving nodes. After all of the energy calculations for the specified number of packet transactions taking place in one 500ms frame were taken into account, the following equation was used to weight the energy costs and calculate network lifetime:

$$Network\ Lifetime = \frac{E_{TOTAL\_CAPACITY}}{\dfrac{E_{transmitter} + E_{receiver} + E_{gateway/Ap} + (N-3)E_{other}}{N * t_{FRAME}}} \tag{5-2}$$

where N is the number of nodes in the network and $E_{TOTAL\_CAPACITY}$ is the total energy capacity of a single node. Each of the protocols used the system model parameters listed in Section 3-7. These analytical models achieved results within 7% of the OPNET Modeler simulation results.

Overall, the percentage difference between the analytical and simulation models averaged to 0.99% for SMAC, 4.38% for TMAC, and 2.64% for GMAC.

5.1.2 IEEE 802.11 Model

The IEEE 802.11 WLAN standard MAC protocol [WLAN97] establishes baseline comparison results for a continuously operating network. This comparison model does not include any power-saving mechanisms. Nodes constantly monitor the medium and do not transition to sleep. For fair comparison, this model implements the same RTS-CTS message reservation mechanisms included in SMAC, TMAC, and GMAC. WLAN networks represented in this model are not constrained by power and designed to sacrifice energy savings for increased throughput and reduced latency. The specific packet sizes used for message exchanges are:

RTS: 13 bytes = 2 byte frame control + 1 byte sequence + 6 byte address (2 src BSS, 2 dst BSS, 1 src, 1 dst) + 2 byte duration + 2 byte FCS

CTS: 13 bytes = 2 byte frame control + 1 byte sequence + 6 byte address (2 src BSS, 2 dst BSS, 1 src, 1 dst) + 2 byte duration + 2 byte FCS

Data MSDU Header and Trailer = 11 bytes = 2 byte frame control + 1 byte sequence + 6 byte address (2 BSS, 1 src, 1 dst) + 2 byte FCS

ACK: 5 bytes = 2 byte frame control + 1 byte sequence number + 2 byte FCS

5.1.3 SMAC Model

The SMAC model reduces idle listening energy consumption by concentrating data transmissions into a smaller, fixed duty cycle. The model achieves a 10% duty cycle by forcing the nodes to sleep for 450ms in a 500ms frame. Unlike the OPNET simulation model, the analytical model does not allow nodes involved in a message exchange to complete the transaction if it extends into the 90% sleep period. The analytical model packet rate remains below the active period saturation point. Incorporating the message passing RTS-CTS techniques to avoid message overhearing, SMAC also saves additional energy by transitioning passive nodes to sleep for the duration of a message intended for another node. The synch messages are not modeled since all of the WSN protocols require these cluster / virtual cluster management mechanisms.

5.1.4 TMAC Model

The TMAC model contains all of the same parameters and functions as SMAC, but also incorporates a variable duty cycle based upon network traffic to reduce idle listening. TMAC sets an adaptive timeout according to the largest possible radio idle period in which a node waiting to transmit could have to wait for a free channel. A hidden transmitting node represents this scenario by conceivably waiting the maximum contention window period before transmitting a RTS. A receiving node would then wait an additional SIFS time before transmitting a CTS. Therefore, the longest idle time for the hidden node before receiving the first portion of the near node's CTS is the maximum contention window time, a RTS message time, and a SIFS time. Using the system model parameters described in Section 3.7 and applying the 1.5 scaling factor produced a TMAC timeout period of 13.48ms.

5.1.5 BMAC Model

The BMAC model uses preamble sampling to periodically wake up and check for traffic. Preamble sampling is designed for ultra-low traffic environments. Choosing a 2.5% duty cycle for a 50 node network from the optimization charts developed in [PoH04] produced a 14ms preamble sampling period for a radio that can perform a clear channel assessment in 0.35ms. A sending node must transmit a preamble which overlaps the entire sampling period in order to ensure the receiver is awake, so the model adds an additional 10% overlap. With random sampling and transmission times beginning throughout the sampling period, a node will average receiving one-half of the extended preamble before the message begins. Like the other protocols, BMAC does not incorporate *early rejection* to immediately drop a packet after receiving the destination field and determining that it is not the intended receiver. In fairness, the low-power listening mechanism for BMAC consumes the same power as the receive mode in all other models.

5.1.6 GMAC Model

The GMAC protocol also uses a 500ms frame time containing a collection period, a GTIM broadcast, and a distribution period. The size of the GTIM is 14 bytes + (3 bytes * number of packets/frame). At the end of both the collection period and the distribution period, the gateway node waits the same 13.48 ms timeout period as TMAC to identify an idle channel before

transitioning to sleep. All other GMAC message exchange mechanisms are the same as described in Chapter 4.

GTIM: 14 bytes = 2 byte frame control + 1 byte sequence + 3 byte address (2 src BSS, 1 dst) + 6 byte absolute time + 2 byte FCS + (n*3 byte) GTIM payload = 14 bytes + 3 * numMsgs

RTS: 13 bytes = 2 byte frame control + 1 byte sequence + 6 byte address (2 src BSS, 2 dst BSS, 1 src, 1 dst) + 2 byte duration in NumOffsets + 2 byte FCS

CTS: 13 bytes = 2 byte frame control + 1 byte sequence + 6 byte address (2 src BSS, 2 dst BSS, 1 src, 1 dst) + 2 byte duration in NumOffsets + 2 byte FCS

ACK: 5 bytes = 2 byte frame control + 1 byte sequence number + 2 byte FCS

SelfNomination: 10 bytes = 2 byte control frame header + 1 byte sequence + 4 byte address (2 src BSS, 1 src, 1 dst) + 1 byte command frame identifier + 2 byte FCS

RegNodeConfirm: 11 bytes = 2 byte control frame header + 1 byte sequence + 4 byte address (2 src BSS, 1 src, 1 dst) + 1 byte command frame identifier + 1 byte winner address payload + 2 byte FCS

GatewayConfirm: 11 bytes = 2 byte control frame header + 1 byte sequence + 4 byte address (2 src BSS, 1 src, 1 dst) + 1 byte command frame identifier + 1 byte winner address payload + 2 byte FCS

Data MSDU Header and Trailer = 11 bytes = 2 byte frame control + 1 byte sequence + 6 byte address (2 src BSS, 1 src, 1 dst) + 2 byte FCS

The NumOffsets duration variable used in the GTIM, RTS, and CTS represents the number of 32µs periods in the transmission exchange. This quantization reduces the size of the field. The election was modeled under the default 6-hour periodic election and scaled appropriately to incorporate the cost into a single 500ms frame period. Although the variety of GMAC control frames appears to place a heavy overhead on the protocol, the 6-hour default election process did not affect the network lifetime.

5.1.7 IEEE 802.11 Infrastructure DCF in Power Save (PS) Model

The IEEE 802.11 DCF in PS mode was modeled with a network consisting of an Access Point (AP) and 49 associated nodes in a manner consistent with the protocol description in section 2.4.2. The beacons with the traffic indication map (TIM) are normally spaced 100ms

apart, but this model spaced them at 500ms to provide an adequate comparison with the other WSN protocols. For simplicity, the TIM is statically set to 8 bytes to represent a traffic bit-map for 64 nodes in the AP's BSS. The standard TIM can range up to 2008 bits. The power save nodes in the network wake up every fourth beacon (i.e. every 2 seconds) to check the TIM and DTIM. All PHY and MAC layer parameters are consistent with the IEEE 802.15.4 models to provide a basis for comparison. The model conducts all download traffic prior to upload traffic to the AP since the PS Poll-SIFS-DATA-SIFS-ACK data download exchange is shorter than the RTS-SIFS-CTS-SIFS-DATA-SIFS-ACK data upload exchange. This exchange order increases the network lifetime performance for the 802.11 DCF in PS mode model.

The RTS and the ACK messages are the same as the WSN frame formats. The 802.11 DCF PS Mode frame formats specific to the model are:

BSS Beacon: 24 bytes = 2 byte frame control + 1 byte sequence + 3 byte address (2 BSS, 1 dst) + 6 byte Timestamp +1 byte DTIM Count + 1 byte DTIM Period + 8 byte TIM (regardless of size of TIM)

PS POLL: 11 bytes = Null Packet = Same as GMAC MSDU header = 2 byte frame control + 1 byte sequence + 6 byte address (2 src BSS, 2 dst BSS, 1 src, 1 dst) + 2 FCS

5.1.8 IEEE 802.11 Ad Hoc Power Save (PS) Model

The IEEE 802.11 ad hoc power save mode was modeled as an independent BSS (IBSS) network consisting of 50 nodes in a manner consistent with the protocol description in Section 2.4.2. The winning TBTT node waits an average contention period, transmits the beacon, and remains awake through the entire frame period. All other nodes listen during the beacon backoff period and transmit ad hoc traffic indication messages (ATIM) during the ATIM window if the nodes have traffic to send. All other nodes listen during the ATIM period. If any ATIM messages are sent, the nodes actively participating in message exchanges remain awake throughout the remainder of the frame to receive packets. If an ATIM message is a broadcast request, all nodes remain awake throughout the following frame. To help the protocol's performance in the comparison, the ATIM window is dynamically set to the time required to send the appropriate number of ATIM requests. Again, all PHY and MAC layer parameters are consistent with the IEEE 802.15.4 models to provide a basis for comparison.

The RTS and the ACK messages are the same as the WSN frame formats. The 802.11 ad hoc PS mode frame formats specific to the model are:

IBSS Beacon: 15 bytes = 2 byte frame control + 1 byte sequence + 3 byte address (2 src IBSS, 1 dst) + 6 byte Timestamp + 1 byte ATIM Window Size + 2 byte FCS

ATIM Request (same as GMAC FRTS): 10 bytes = 2 byte frame control + 1 byte sequence + 4 byte address (2 src IBSS, 1 src, 1 dst) + 1 byte duration in NumOffsets + 2 FCS

5.2 Protocol Simulation Models

The GMAC simulation evaluation required implementing GMAC, SMAC, and TMAC in OPNET and using an object-oriented approach to vary packet, network, and traffic sizes to analyze their performance. The purpose of the GMAC and RPM algorithm simulations is to sufficiently address the research questions posed in Section 1.4.

Each of the WSN MAC protocols was implemented in OPNET Modeler using a top-down design approach to evaluate and compare the protocols using the network performance metrics. The hierarchical structure of network scenarios, nodes, and processes provides a comprehensive developmental environment to model WLAN and WPAN networks. The Discrete Event Simulation (DES) tool in the OPNET Modeler has a fully developed IEEE 802.11 wireless package to modify in order to model both the behavior and performance of GMAC.

The OPNET WSN MAC protocol model developed in this research contains many of the original WLAN IEEE 802.11 functions. The OPNET Modeler standard 802.11 model included the basic packet flow functions needed to model WSN DCF protocols. The most significant WSN modifications to this model were the creation of a new sleep state within the MAC finite state machine (FSM), the modification of the message passing NAV calculations, the adjustment of the MAC service data unit (MSDU) and physical layer convergence procedure (PLCP) message formats, the development of an energy consumption model, and the implementation of a radio power management algorithm. These modifications provide the framework to develop and evaluate the performance of various WSN protocols in OPNET.

5.2.1 Top-down Design

The OPNET Modeler Discrete Event Simulator provides a top-down network design environment which eases the development of WSN MAC protocols. The top-level network shown in Figure 5-1 depicts the random placement of 50 nodes in a 100m by 100m bounded network. Each node implements a node model with its associated reduced protocol stack shown in Figure 5-2. The boxes in the node models each represent a finite state machine (FSM). The left side shows how the GMAC protocol builds upon the IEEE 802.15.4 MAC layer to perform the additional data collection and distribution functions. The right side of the figure shows the SMAC node implementation. OPNET allows the abstraction of the top network layers into simple source and sink models for MAC- and PHY-centered simulation models. Figures 5-3 and 5-4 illustrate the finite state machine (FSM) implementations for the GMAC and WSN MAC layers, respectively.

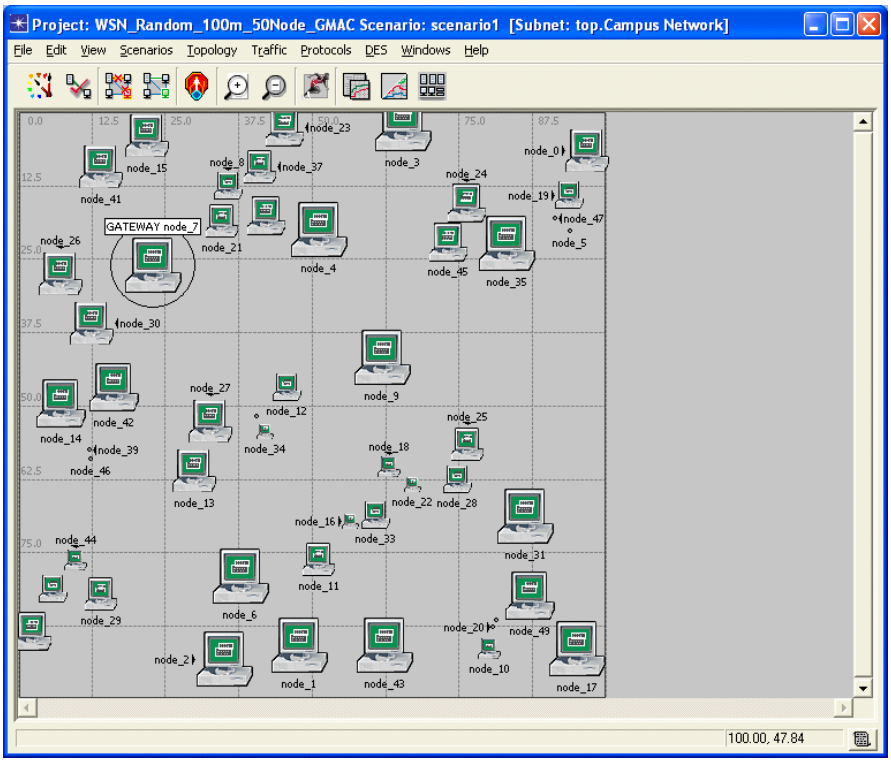

Figure 5-1 OPNET 50 Node 100mx100m Random Scenario

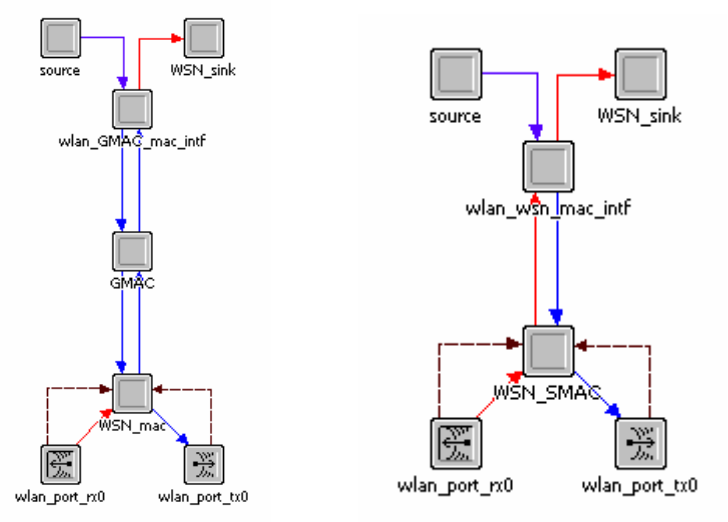

Figure 5-2 OPNET GMAC (left) and SMAC (right) Node Models

5.2.2 WSN Channel Contention

The slotted, contention-based WSN protocols require modifying the standard IEE 802.11 WLAN contention mechanism to account for data queued during the sleep periods. Standard WLAN channel contention does not require nodes to perform an exponential backoff for the first medium access attempt if the DIFS deference period does not indicate a busy channel. This medium access control policy is effective in WLAN applications because message transmission requests occur in an asynchronous manner. WSN applications can also generate messages in an asynchronous manner; however, WSNs accumulate these messages during sleep periods and attempt to simultaneously send them on the first time slot edge of the active period. Employing a similar policy of not requiring an initial backoff would result in all WSN nodes waiting the initial DIFS deference period and then simultaneously attempting to send their traffic. The IEEE 802.11 channel access mechanism was modified to require *all* channel contention attempts to perform an exponential backoff and prevent the costly frame collisions.

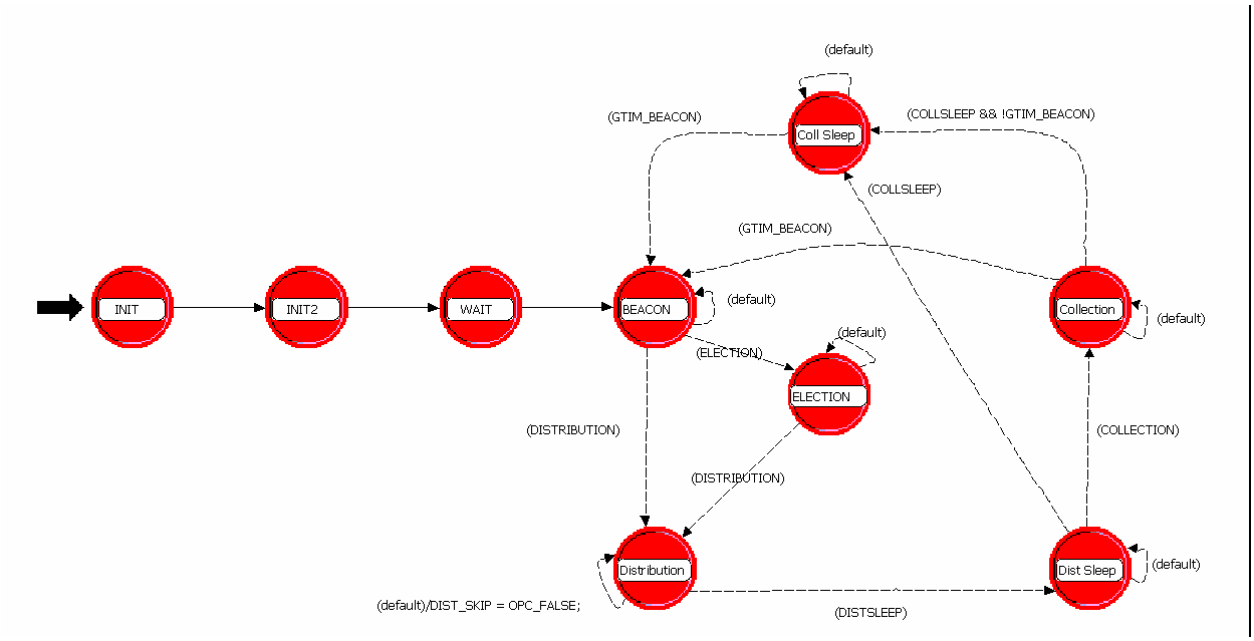

Figure 5-3 GMAC Process Model

5.2.3 WSN MAC Sleep State Design

The MAC layer SLEEP state shown in Figure 5-4 models WSN protocols which transition their radios to sleep in order to conserve energy. Implementing the node sleep mode as an explicit state within the MAC layer FSM instead of the function block (FB) eliminated the requirement to provide sleep management mechanisms in each of the other MAC layer states where the node could be operating during the initiation of sleep intervals. Individual SMAC nodes control their sleep state by scheduling appropriate sleep and wakeup self-interrupts. When the sleep interrupt occurs, a service routine switches nodes to the sleep mode if their current state allows them to immediately transition to sleep. However, if two nodes start a RTS-CTS sequence to initiate a data transmission before their regular sleep interrupt time, the nodes will remain awake to finish their transmission and then transition to sleep. Nodes store their pre-sleep state information and resume their regular activities upon waking. To maintain network synchronization and efficient network operation, nodes must offset pending deference and contention backoff self-interrupts by the sleep duration. Additionally, NAV durations may require resetting if an active transmission carried into a regular sleep cycle.

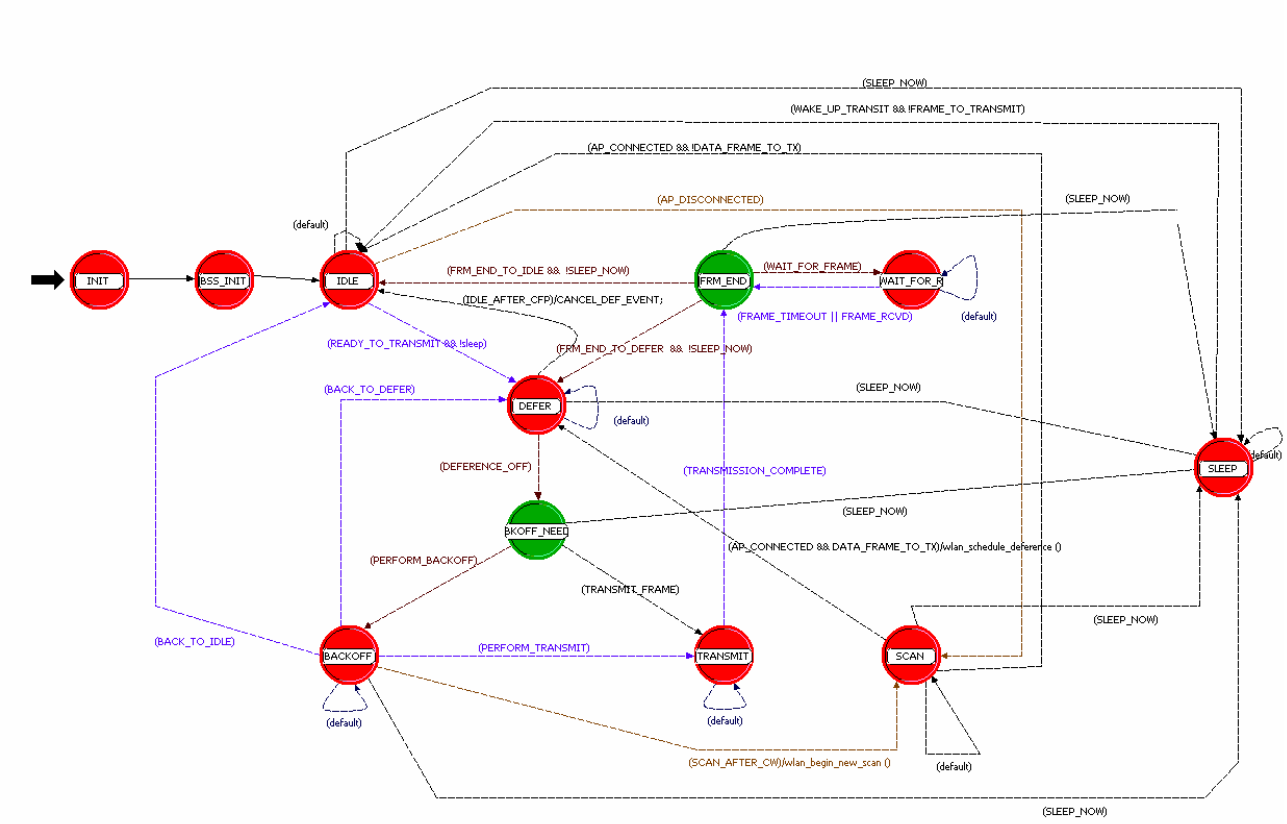

Figure 5-4 WSN MAC Process Model with Sleep State

TMAC nodes set an adaptive timeout (TA) interrupt and transition to sleep if the channel remains idle for the duration of the interrupt. Any network activity cancels the TMAC TA interrupt, and the interrupt is reset when the channel returns to idle. A recurring 500ms interrupt automatically wakes up all sleeping TMAC nodes.

All packet transmissions are initiated with an RTS-CTS sequence to avoid frame collisions and allow nodes to sleep during data transmissions not intended for them (message overhearing). Since low active duty cycles concentrate all network data transmissions into a smaller active period, a RTS-CTS sequence reserves the channel before each data transmission and lowers the probability of frame collisions. RTS messages also create extended sleep opportunities while implementing message passing because nodes sleep during transmission sequences not intended for them. Upon receiving a RTS, a node evaluates the destination field. If not the intended receiver, the node extracts the duration of the remaining transmission exchanges, schedules an immediate sleep interrupt for the projected completion time, and transitions to a sleep mode

called NAV sleep. In order to simplify NAV sleep, stream interrupts are not disabled in the sleep state, and nodes automatically discard any packets received while asleep.

The two OPNET simulation output graphs in Figure 5-5 illustrate a single node's activity for a five second network period with a 30% duty cycle. The extended frame period and active cycle are not typical SMAC values, but the figures illustrate the effect of NAV sleep. The values on each graph are high when the radio is in active receive mode and low when the node is asleep. The left picture in Figure 5-5 depicts a complete active and sleep cycle, and the right picture highlights the active period effects of NAV sleep in the network.

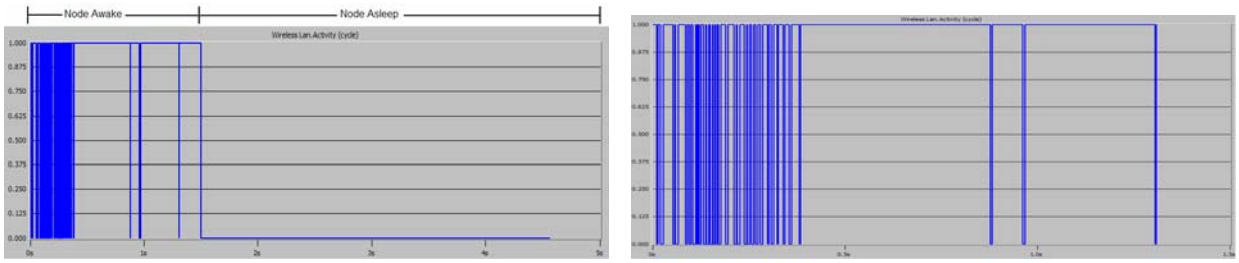

Figure 5-5 S-MAC node period (left) and NAV Sleep Active Period (right)

5.2.4 WSN MAC Message Passing

Implementing message passing to transmit more data and permit nodes to sleep for the longest possible duration required the modification of the NAV table calculations. In message passing, nodes are allowed to send multiple packets to a single destination after gaining contention of the medium by successively transmitting the packets and receiving their respective ACKs. The OPNET IEEE 802.11 model provides the ability to set a fragmentation threshold and adjust the MSDU maximum size to permit larger messages, but the IEEE 802.11 standard sets the duration period to accommodate only one fragment transmission sequence. Extending the duration across the entire fragmentation sequence permits nodes to sleep longer in a lower power-saving mode. The formulas used to calculate the message passing packet duration field are:

$$numFrags = CEILING(messageSize / fragmentationThreshold) \quad (5\text{-}3)$$

$$\text{durationField} = \text{SIFS}+\text{CTS}+ \text{numFrags} * (\text{SIFS}+\text{MSDU Header}+ \text{fragmentationThreshold}+ \text{SIFS} + \text{ACK}) \quad (5\text{-}4)$$

Another aspect of message passing is NAV sleeping. As previously discussed, nodes can go to sleep after verifying that a RTS or CTS message is not intended for them. Nodes will set NAV duration based upon the duration field and determine if enough time exists to transition to and recover from sleep. If the RPM algorithm is disabled, then nodes can only go to sleep if the NAV duration is longer than the LPM3 transition requirement. However, if the RPM algorithm is enabled, nodes will determine the optimum level of sleep depending on the NAV duration and the threshold time of each LPM level.

5.2.5 WSN MAC Energy Consumption Calculations

Developing an energy consumption model to measure protocol energy-efficiency performance requires the calculation of the time spent in each of the three general radio states in the FSM —IDLE, TRANSMIT, and SLEEP. Any state other than TRANSMIT or SLEEP is categorized as IDLE and charged the energy consumption rate for the radio receive mode. Upon transitioning out of forced states in the MAC layer FSM, the state exit executives check to see if the next state applies a different energy consumption rate. For instance, nodes transitioning from IDLE state to DEFERENCE state would not require an energy calculation, but nodes transitioning from BACKOFF state to TRANSMIT state would require an energy calculation. The energy update calculation uses the state duration time and the associated energy consumption rate to calculate the energy consumed. The accumulating energy consumption is compared with the battery capacity to determine the node lifetime. Once a node consumes the same amount of energy as the battery capacity, the node is deregistered to remove it from the list of available receivers and placed in a permanent sleep mode.

5.3 GMAC State Design Descriptions

The GMAC OPNET process model was designed using a top-down approach to decompose GMAC into six distinct phases and to link them together using a finite state machine (FSM). Figure 5-6 illustrates the highest level of abstraction for the GMAC protocol FSM. Although the state flow diagrams described in Appendix A detail an election algorithm to self-

configure and self-recover a cluster, the simulation model initializes one node as the gateway and all others as regular nodes. All nodes enter the GTIM beacon state and immediately synchronize the network with the first transmitted empty GTIM. This section provides a general overview on the functionality of each GTIM state, and Appendix A provides a more detailed flow chart description.

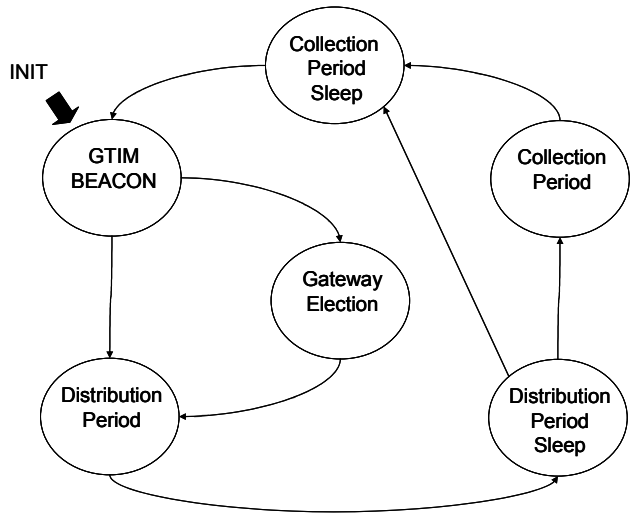

Figure 5-6 GMAC Finite State Machine Design

5.3.1 GTIM Beacon Period

Nodes begin the GMAC process cycle by entering the GTIM beacon period and waiting for a maximum of 3 cycles to receive a GTIM beacon. The GTIM beacon period state process is shown in Figure 5-7. During a normal cycle, the regular nodes wake up one SIFS period earlier than the gateway to reduce the required synchronization precision and to ensure that they receive the entire GTIM beacon. The gateway node then enters the GTIM Beacon state and decides whether to signal for a new gateway election based upon its resource levels or the default six-hour gateway rotation counter. If the cycle requires a new gateway election, the gateway transitions to the election state and sends a GTIM with the election flag set and the next distribution schedule attached. All regular nodes receive the election GTIM and transition to the election state. If the cycle is not an election cycle, the gateway builds the schedule and transmits the non-election GTIM. A PHY-layer timestamp ensures that the network 6-byte timestamp is

accurate. Once all nodes receive the non-election GTIM, they simultaneously transition to the distribution period.

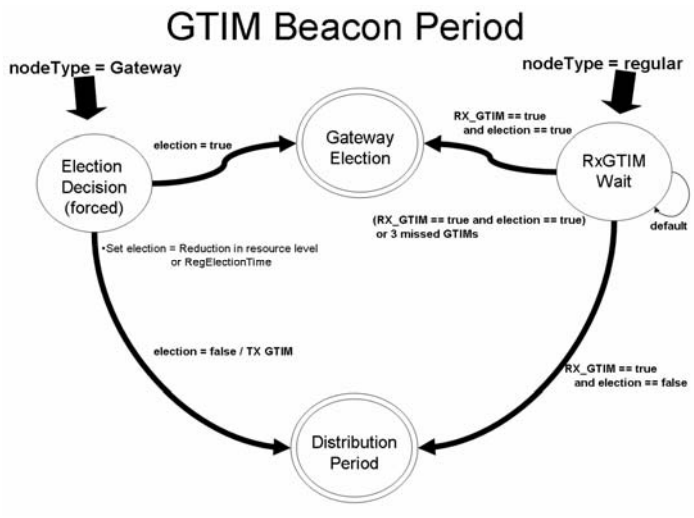

Figure 5-7 GTIM Beacon State

5.3.2 Distribution Period

Upon entering the distribution period shown in Figure 5-8, all nodes immediately begin parsing the GTIM schedule and accumulating sleep time until their first transaction. The iterative parsing algorithm in Appendix D details an interrupt-driven method to accumulate sleep and then suspend parsing until the next scheduled transaction time. Setting sleep and parsing resumption interrupts reduces the complexity of the scheduling algorithm on processor- and memory-challenged WSN platforms. Once nodes iterate through the entire schedule, they transition to the distribution sleep state. Nodes send and receive data in the scheduled TDMA slots. This contention-free distribution state offers enhanced network stability and performance under heavy loads due to the scheduled TDMA nature.

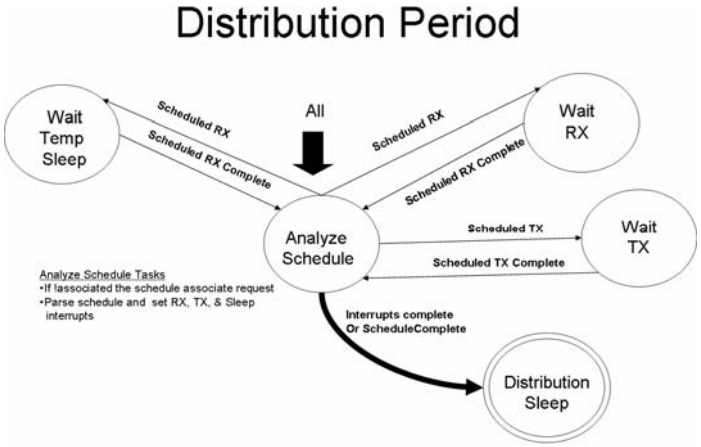

Figure 5-8 Distribution State

5.3.3 Distribution Sleep Period

When a node has completed all of its scheduled distribution data exchanges, the node transitions to the distribution sleep state and sets two interrupt signals: a sleep interrupt to allow the WSN MAC layer to go to a low power mode and a GMAC collection period interrupt to transition to the collection period. The WSN MAC process is set to sleep all the way through the collection period to preclude an unnecessary radio wakeup energy cost if it does not have queued messages at the beginning of the collection period. The sleep interrupt can be cancelled at any time. As shown in Figure 5-9, the gateway node enters the distribution sleep state and sets an adaptive timeout to begin at the end of all scheduled transactions for inter-network and sleep. In another dynamic windowing version of GMAC, the collection period begins immediately upon completion of the distribution schedule to utilize the bandwidth more efficiently.

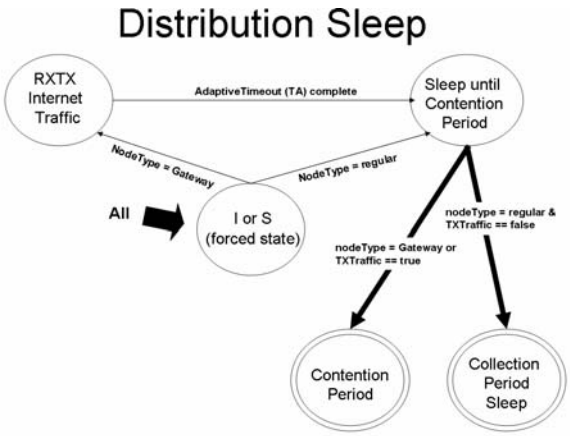

Figure 5-9 Distribution Sleep State

5.3.4 Collection Period

Nodes with data queued for transmission cancel the WSN MAC low-power mode (LPM) sleep with an early wakeup function call to provide the PHY layer sufficient time to power up the radio subsystems. Next, nodes compete for medium access using standard contention-based DIFS deference and exponential backoff to reserve distribution period time slots with the gateway via future request to send (FRTS) control messages. The random exponential backoff guarantees fair competition for the distribution period scheduled slots.

Figure 5-10 shows that the gateway immediately sets an adaptive timeout when it arrives in the collection state. A sender transmits a FRTS message to the gateway indicating the message destination address and duration. The gateway acknowledges the request and immediately adds the transaction onto the tail of the GTIM schedule list. Once a node successfully requests a reservation for all of its queued messages or receives a message from the gateway stating that the schedule is full, the node transitions to the collection sleep state. When the gateway adaptive timeout is complete and it has transmitted any queued inter-network packets, the gateway remains in the collection period with the WSN MAC process in a low power mode until the GTIM beacon time.

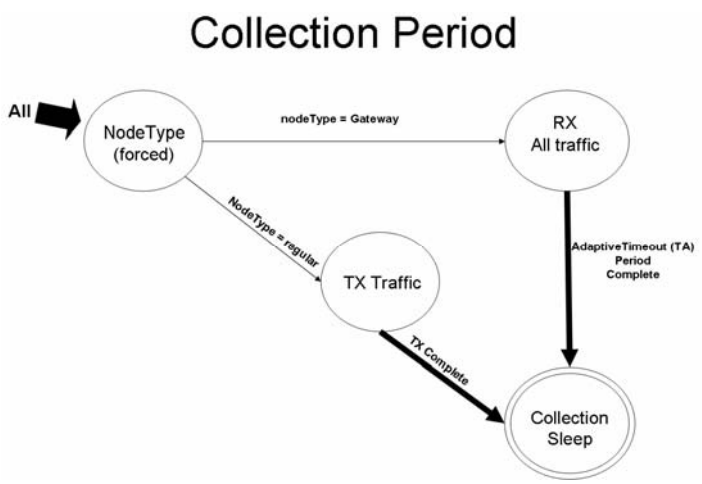

Figure 5-10 Collection State

5.3.5 Collection Sleep Period

GMAC extends network lifetime by allowing nodes to sleep during the collection period when they have no queued data to transmit. The nodes set a WSN MAC process wakeup interrupt and GMAC GTIM beacon state transition interrupt to restart the entire cycle at the next GTIM beacon time. Figure 5-11 illustrates the collection sleep process.

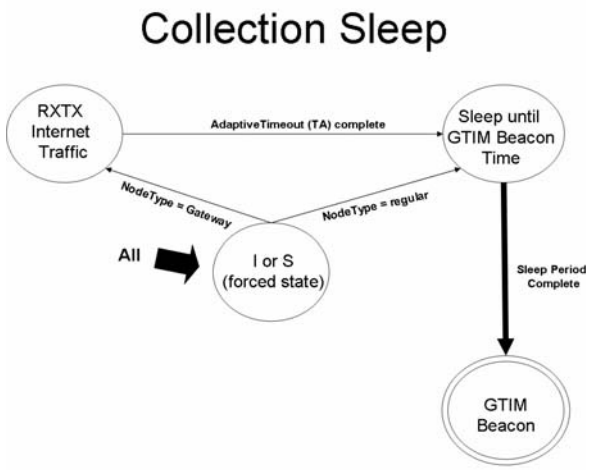

Figure 5-11 Collection Sleep State

5.3.6 Gateway Election: RAVE Algorithm

The resource adaptive voluntary election (RAVE) algorithm allows for node self-election based upon each node's available battery and memory resources. GMAC periodically elects a new gateway node to equally distribute the energy requirements among all of the nodes using the RAVE scheme. As described in the GTIM beacon state, the gateway evaluates the need for an election and transitions to the gateway election state. The gateway sets the election flag field and sends a GTIM beacon to the regular nodes still idly waiting in the GTIM beacon state. Figure 5-12 shows how the gateway node sets an election timer and waits for nodes to contend to send a self-nomination message based upon the passive resource-based RAVE algorithm described in Section 4.3.2. Figure 5-13 and the election state flow diagrams in Appendix A detail how the regular nodes listen for a GTIM or other self-nominations before transmitting their own self-nomination message. The gateway returns a gateway confirmation message when it successfully receives a self-nomination message. Finally, all nodes set an interrupt for the next GTIM beacon

time and immediately transition to the distribution state to parse the schedule sent in the original election GTIM message. All of the schedule offset times are relative to the beginning of the distribution period, so election timing is not critical. The address of the gateway is statically reserved as 254 (or -2 in OPNET), so if a node does not need to receive the winning gateway node's address to recover or join the network.

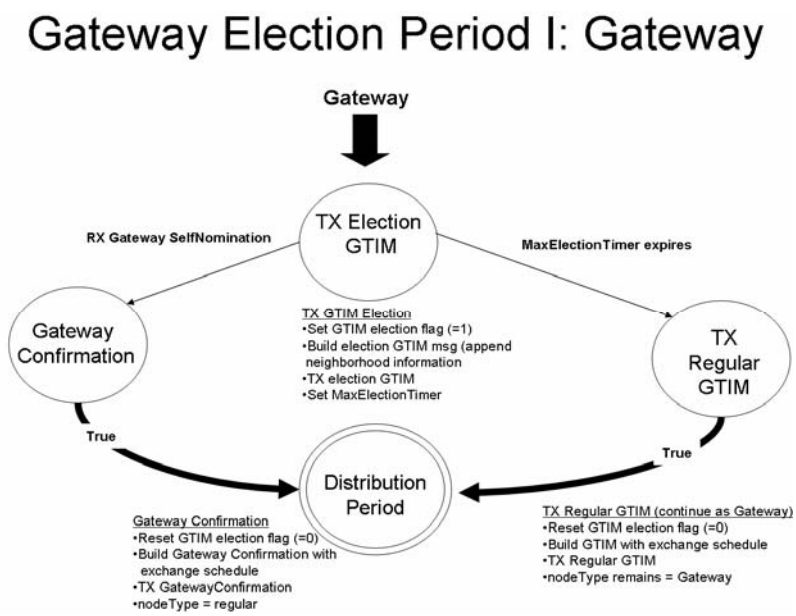

Figure 5-12 Gateway Election State Gateway Node

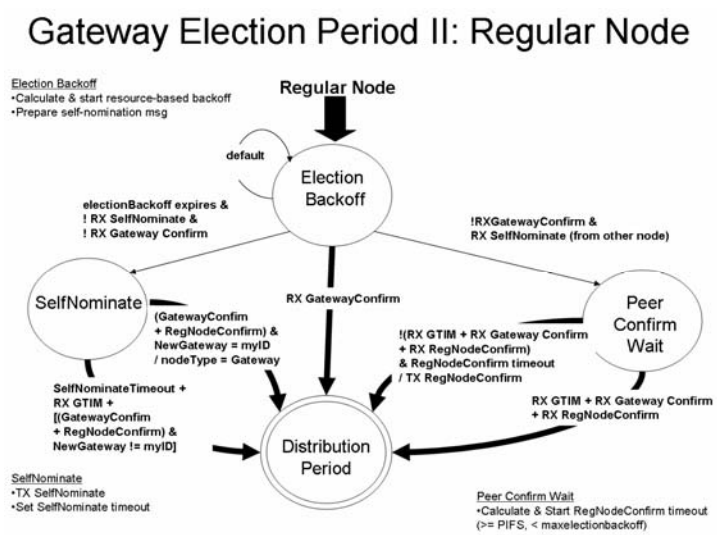

Figure 5-13 Regular Node Election State

The election state also handles the special circumstances when a cluster is first initialized or the current gateway fails. These unique cases are detailed in Appendix A.

5.4 Summary

This chapter has presented the analytical and simulation models to analyze the WSN protocols using the network performance metrics. Section 5.1 described the analytical models used to verify the OPNET Modeler 11.0 simulation models. Section 5.2 described the top-down OPNET Modeler system design and the primary contention, sleep, and power mechanisms used to model the GMAC, TMAC, and SMAC protocols. The simulation models modified the IEEE 802.11 WLAN protocol standard to implement IEEE 802.15.4 wireless sensor network devices and the corresponding energy-efficient protocol mechanisms. This new WSN model not only provides a means to evaluate the performance of WSN MAC protocols, but the model also provides a simulation tool to evaluate their associated hardware radio and mote platforms. Section 5.3 described the OPNET Modeler implementation of the GMAC process finite state machine. The next chapter uses the models described in this chapter to analyze the GMAC protocol performance.

Chapter 6

GMAC Protocol Performance

The only source of knowledge is experience.
--Albert Einstein

This chapter presents the research results and analysis for the GMAC protocol. Section 6.1 establishes the statistical accuracy of the data. Each protocol is evaluated in sections 6.2 and 6.3 using the three primary performance metrics: network lifetime, sleep percentage, and energy consumed per data bit. Additional analysis shows that the centralized management architecture in GMAC provides increased security measures against denial of sleep attacks. Overall, Chapter 6 focuses on developing several of the answers to the research questions posed in Section 1.4:

1. What are the benefits of Gateway MAC?
    a. In terms of energy savings
    b. In terms of extended network lifetime
2. What are the latency costs of GMAC?
3. What are the security vulnerabilities of GMAC and how can they be mitigated?

6.1 Statistical Accuracy

Taking samples of systems which invoke stochastic processes introduces a measure of uncertainty in the results. Many discrete event simulation packages use random number generator seeds to calculate random numbers and to allow those numbers to be repeatable for future simulations. Simulating a scenario with one seed provides a valid result for that scenario, and selecting 5 to 10 random seed sets for the same scenario decreases the confidence interval [Mac87]. This research uses five randomly selected seeds to generate simulation results for each scenario. From these results, the 95% confidence interval is calculated to determine the statistical accuracy of the simulations. The 95% confidence interval (CI) provides the two probabilistic bounds, c1 and c2, in which there is a (1- α) probability that the population, or actual, mean is in that interval (c1, c2):

$$\text{Probability } \{c1 \leq \mu \leq c2\} = 1 - \alpha \tag{6.1}$$

where μ is the population mean and α is the significance interval (0.05 for the 95% CI case [Jai91]. Since the sample size is not large, the CI cannot be determined using the standard Central Limit Theorem. An alternative method for small sample sizes which are derived from a normally distributed population determines the 100 * (1- α) confidence interval using:

$$\bar{\chi} - t_{\left[1-\frac{\alpha}{2}, n-1\right]} \frac{s}{\sqrt{n}}, \bar{\chi} + t_{\left[1-\frac{\alpha}{2}, n-1\right]} \frac{s}{\sqrt{n}} \tag{6.2}$$

Where $\bar{\chi}$ is the mean, s is the standard deviation, n is the number of samples. $t_{[1-\alpha/2, n-1]}$ represents the ($1-\frac{\alpha}{2}$) –quantile of a t-variant with n-1 degrees of freedom. For a 95% CI and sample size n = 5 random seeds, the t distribution $t_{[1-\alpha/2, n-1]}$= 2.776.

The confidence intervals for the GMAC network lifetime simulation results in Table 6-1 show that the GMAC network lifetime had a tight confidence interval. For example, the left column illustrates the network lifetime for a GMAC network with 20 packets per second. The average network lifetime for each of the random seeds resulted in a mean value of 367.64 days

and a 95% confidence interval ranging from 367.09 to 368.11. The largest difference between the maximum and minimum lifetime for a given packet rate was 0.3%.

Table 6-1 95% Confidence Interval for GMAC Network Lifetime (days) vs. Network Packet Rate

	20 pkts/s	16 pkts/s	12 pkts/s	10 pkts/s	8 pkts/s	6 pkts/s	4 pkts/s	2 pkts/s	0 pkts/s
seed 1	368.1	438.2	529.9	587.5	654.6	736.7	833.5	942.9	1025.6
seed 2	367.6	438.2	529.9	586.6	654.1	736.1	834.8	942.5	1025.6
seed 3	367.0	438.3	529.5	586.9	655.4	736.4	833.7	942.5	1025.6
seed 4	367.5	438.5	529.6	587.0	654.6	737.3	833.9	942.9	1025.6
seed 5	368.0	438.0	529.6	587.0	655.2	736.5	834.6	942.2	1025.6
Mean	367.64	438.24	529.70	587.00	654.78	736.60	834.10	942.60	1025.60
std dev.	0.4393	0.1817	0.1871	0.3240	0.5215	0.4472	0.5701	0.3000	0.0000
variance	0.1930	0.0330	0.0350	0.1050	0.2720	0.2000	0.3250	0.0900	0.0000
95% C1	367.09	438.01	529.47	586.60	654.13	736.04	833.39	942.23	1025.60
95% C2	368.11	438.48	529.98	587.46	655.34	737.26	834.69	942.99	1025.60

The confidence intervals for the TMAC network lifetime simulation results in Table 6-2 show that the TMAC network lifetime also had a tight confidence interval. For example, the left column illustrates the network lifetime for a TMAC network with 20 packets per second. The average network lifetime for each of the random seeds resulted in a mean value of 39.8 days and a 95% confidence interval ranging from 39.77 to 39.91. The largest difference between the maximum and minimum lifetime for a given packet rate was 1.75%. The results for all simulations exhibit similar 95% confidence intervals.

Table 6-2 95% Confidence Interval for TMAC Network Lifetime (days) vs. Network Packet Rate

	20 pkts/s	16 pkts/s	12 pkts/s	10 pkts/s	8 pkts/s	6 pkts/s	4 pkts/s	2 pkts/s	0 pkts/s
seed 1	39.9	47.8	58.8	66.2	76.9	90.8	110.0	140.1	194.3
seed 2	39.9	47.6	58.7	66.1	76.5	89.9	110.2	141.0	194.3
seed 3	39.8	47.5	58.7	66.5	76.7	90.5	110.4	142.6	194.3
seed 4	39.8	47.6	58.7	66.6	77.1	90.7	109.9	140.6	194.3
seed 5	39.8	47.3	58.4	66.1	76.9	90.2	110.1	141.2	194.3
Mean	39.84	47.56	58.66	66.30	76.82	90.42	110.12	141.10	194.30
std dev.	0.0548	0.1817	0.1517	0.2345	0.2280	0.3701	0.1924	0.9381	0.0000
variance	0.0030	0.0330	0.0230	0.0550	0.0520	0.1370	0.0370	0.8800	0.0000
95% C1	39.77	47.33	58.47	66.01	76.54	89.96	109.88	139.94	194.30
95% C2	39.91	47.79	58.85	66.59	77.10	90.88	110.36	142.26	194.30

6.2 WSN Model Comparisons

The Sensor MAC, Timeout MAC, and Gateway MAC WSN MAC protocols were modeled in OPNET Modeler as described in the previous chapter. This section evaluates and compares each of the WSN protocols varying by the simulation factors and observing the performance metrics.

Results from the WSN energy-efficient MAC protocol OPNET simulations show that GMAC significantly outperformed SMAC and TMAC in maximizing sleep percentage, extending network lifetime, and minimizing energy consumption per data bit under all traffic conditions. All three protocols have similar throughput capabilities, but SMAC must statically set the active duty cycle upon deployment. TMAC and GMAC dynamically adjust the active duty cycles to accommodate the current traffic conditions. Finally, GMAC is able to maintain stable end-to-end inter-network traffic delays, but the protocol creates more than twice the end-to-end data delay of the other protocols for intra-network data exchanges using the data collection and distribution periods. This increased end-to-end delivery delay is the performance metric that is sacrificed to provide GMAC the ability to increase node sleep durations and significantly extend network lifetime.

6.2.1 WSN Sleep Percentage vs. Unicast Traffic Packet Rate

The first simulation performance metric to analyze is the average node sleep percentage for various network unicast packet rates. The percentage sleep metric provides the most significant insight into the subsequent network lifetime and energy/bit results. The experimental measurements detailed in Chapter 7 show that the Tmote Sky WSN platform LPM3 sleep level draws an average current of 38 μA, while the receive and transmit modes draw 21.6 mA and 18.4 mA, respectively. With typical 3000 mAh lithium batteries, the difference in lifetime of a fully active Tmote platform (21.6mA) and a sleeping platform (38 μA) is 5.8 days vs. 9.0 years (or battery shelf life). Enhancing a protocol to extend the average node sleep percentage from 90% to 91% would increase the node lifetime from 57 days to 63 days. Therefore, a 1% increase in sleep produced a 10% increase in node lifetime.

Table 6-3 and Figure 6-1 show that the GMAC data collection and distribution scheme provides significantly longer sleep opportunities than the DCF SMAC and TMAC schemes. The

bottom row in Table 6-3 represents the network lifetime percentage increase of the GMAC protocol over the TMAC protocol. The sleep percentage results obtained in the OPNET Modeler simulations confirm the expectations of qualitative protocol analysis conducted in Chapter 3. The simulation model parameters set the static SMAC active duty cycle to 10%. This 10% duty cycle limits the available transmission window to 100 ms out of every second and reduces the available bandwidth to support a throughput of only 20 packets/sec. If the application using the SMAC protocol requires a higher throughput, the active duty cycle must be manually adjusted prior to deployment. The slight increase in SMAC sleep percentage with higher packet rates is due to increased NAV sleep opportunities. With no traffic, all nodes monitor the idle medium in receive mode throughout the 10% active cycle. NAV sleep allows nodes which are not actively participating in the message exchange to sleep after receiving an RTS not intended for them and extracting the duration field. This increased sleep percentage is slightly reduced by communicating nodes extending their active cycle when completing an exchange that was in progress at the beginning of the SMAC sleep cycle interrupt. This slight reduction is observable in *No NAV* simulation scenarios presented in Chapter 7. TMAC displays a monotonically decreasing sleep percentage slope as the packet rate increases. Unless the network saturates, each TMAC node waits a fixed 13.48 ms TA timeout in the receive mode. With a 500 ms frame, the TA adds a 2.70% idle listening active period to all network packet rate scenarios. The results shown in Table 6.3 clearly illustrate this effect in the 0 packet/sec case in which TMAC sleeps 97.3% of the time. The steep sleep percentage slope in TMAC is caused by the network-wide overhearing cost for each packet sent in the network. GMAC, however, only adds a network-wide 3-byte increase in the GTIM size for each additional packet. Like the TMAC adaptive timeout frame overhead, GMAC also expends a base rate for the GTIM. As described in Chapter 5, The GTIM packet size is 14 bytes + (3 bytes * number of packets in the schedule) + SIFS early wakeup robustness. At a transmission rate of 32 μs/byte and an equivalent 6-byte PHY header, these packet sizes correspond to a network-wide listening cost of 640 μs + 192 μs for 0 packet/s (0.166% of frame), 1.12 ms + 192 μs for 10 packets/s (0.262% of frame), and 1.59 ms + 192 μs for 20 packets/s (0.357% of frame) for each frame period. In GMAC, only the gateway node remains awake for a 13.48 ms TA timeout during both the collection time and the distribution time for a combined 26.96 ms (5.39% of frame). With 50 nodes effectively sharing this cost through the rotating gateway duties, the average cost per node in 0 packet/sec traffic

conditions is 0.166% + 5.39% / 50 nodes = 0.286% average active period. Table 6.3 illustrates this effect in the 0 packet/s case where GMAC nodes sleep an average of 99.7% of the time. The decreasing slope in sleep percentage for GMAC as packet rates increase is smaller than the TMAC slope since each additional packet only adds a small number of active nodes to overhear the reservation process (gateway and nodes with unacknowledged data to transmit). Furthermore, the actual added data exchange only requires the data sender and receiver to be awake.

Table 6-3 WSN Ave Sleep Percentage vs. Network Unicast Packet Rate

	Packet Rate							
Protocol	60 pkt/s	40 pkt/s	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	0 pkt/s
SMAC	N/A	N/A	90.3%	90.2%	90.1%	90.0%	90.0%	90.0%
TMAC	56.1%	73.7%	85.6%	88.0%	90.3%	92.6%	94.9%	97.3%
GMAC	94.9%	97.1%	98.7%	99.0%	99.2%	99.4%	99.6%	99.7%
% GMAC Increase	53.4%	31.7%	15.4%	12.5%	9.9%	7.3%	4.9%	2.5%

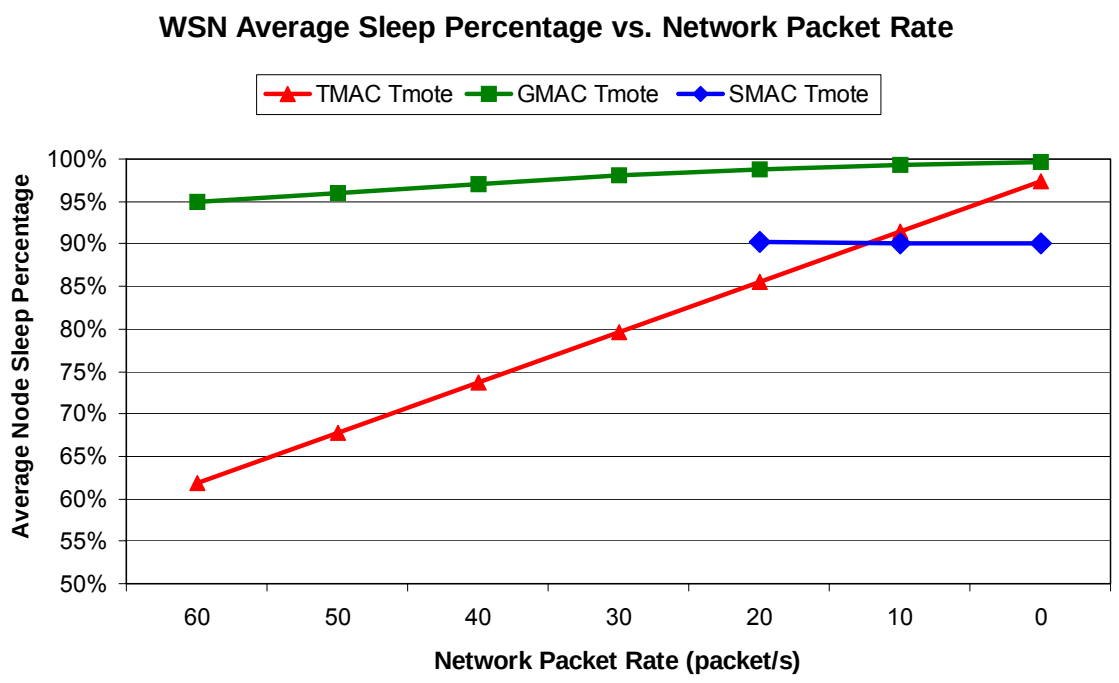

Figure 6-1 WSN Sleep Percentage Network Lifetime vs. Network Unicast Packet Rate

As the traffic rate increases to full saturation, SMAC will remain at a 90% sleep percentage and drop any packets beyond the available bandwidth. With full bandwidth utilization, TMAC will approach 0% sleep percentage and GMAC will approach 70% sleep percentage. Since GMAC network-wide listening can occur only during the collection period, the protocol sleep percentage will not approach 0%. The ratio of collection to distribution period for the largest packet size is 27.7% collection period to 72.3% distribution period, or (Contention + DIFS + FRTS + SIFS + ACK) to (SIFS + DATA + SIFS + ACK). Assuming that the nodes remain awake throughout the collection period, receive the GTIM with a full schedule, receive their individual share of the maximum-sized packets, and sleep the remainder of the distributed packets, each node will still be able to sleep 70% of the time in a saturated network. The estimated lifetime would be 19.2 days. The TMAC estimated lifetime in a fully saturated network is 5.7 days.

These average node sleep percentage results verify that GMAC effectively eliminates network-wide idle listening and significantly reduces message overhearing by having the gateway node monitor the network for traffic and dynamically establish a schedule while the other nodes sleep. The analysis describing these sleep percentage results also applies to the remaining performance metrics.

6.2.2 Network Lifetime vs. Unicast Traffic Packet Rate

The OPNET Modeler simulation results for the network lifetime performance metric while varying the unicast packet rate directly follow from the comparison with the sleep percentage metric. Table 6-4 and Figure 6-2 show that GMAC outperforms the other protocols in network lifetime for all of the unicast packet rates. With the nodes consuming almost three orders of magnitude more energy in the active than in the sleep mode, a small increase in sleep percentage has a tremendous impact on network lifetime. For example, GMAC sleeps 2.4 % longer than TMAC under empty network traffic conditions which creates a 428% increase in network lifetime from 194 days to 1026 days. The bottom row in Table 6-4 represents the network lifetime percentage increase of the GMAC protocol over TMAC protocol. SMAC is guaranteed to last at least 56.4 days with its fixed 90% duty cycle. The TMAC network lifetime will range from 5.7 days with a saturated 180 packet/s network to 194 days with an empty traffic network.

Finally, the GMAC network lifetime will range from 19.2 days with a saturated 180 packet/s network to 1026 days with an empty traffic network. Since these ultra-low powered WSN networks are designed to average packet rates on the order of 6 to 24 seconds per packet (0.042 pkts/s to 0.166 pkts/s), the true application lifetime will closely resemble the 0 pkt/sec simulation results. As with the average sleep percentage, GMAC is able to significantly increase the network lifetime across all packet rates.

Table 6-4 WSN Network Lifetime vs. Network Unicast Packet Rate

	Packet Rate							
Protocol	60 pkt/s	40 pkt/s	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	0 pkt/s
SMAC	N/A	N/A	58.3 days	57.7 days	57.0 days	56.6 days	56.5 days	56.4 days
TMAC	15.3 days	22.0 days	39.4 days	47.6 days	58.7 days	76.8 days	110 days	194 days
GMAC	106 days	179 days	368 days	438 days	530 days	655 days	834 days	1026 days
% GMAC Increase	593%	650%	823%	821%	803%	752%	657%	428%

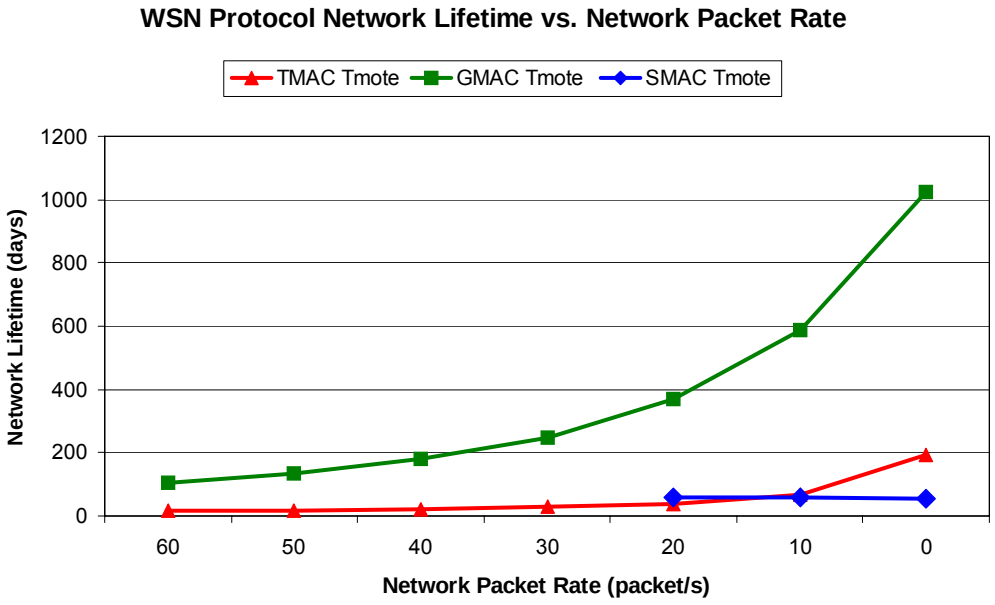

Figure 6-2 WSN Network Lifetime vs. Network Unicast Packet Rate

6.2.3 Network Energy/Bit vs. Unicast Traffic Packet Rate

The OPNET Modeler simulation results for the network energy per data bit lifetime performance metric while varying the unicast packet rate are consistent with the comparison descriptions of the sleep percentage metric. Network energy/bit provides insight beyond the sleep percentage metric. This metric also reveals the efficiency of a protocol in sending data information. The OPNET scenarios accumulated the total number of data bits delivered over the lifetime of the network. The total network energy consumed to the network total delivered data bits ratios are shown in Figure 6-3 and Table 6-5. Factors that affect this variable include all of the standard sources of WSN energy loss: idle listening, frame collisions, message overhearing, and protocol overhead (MSDU header overhead data bits and control packets). Each of the WSN MAC protocols have increased frame collisions due to compressing data exchanges into smaller active periods, and they mitigate the frame collision effects by sending small reservation packets and implementing standard exponential backoff procedures. GMAC also limits idle listening to the gateway node and any node contending to request a schedule reservation. With an FRTS-ACK reservation scheme which is similar to the TMAC and SMAC RTS-CTS reservation process, GMAC is able to schedule exclusive data exchange slots and eliminate network-wide message overhearing. With gateway elections and rotations scheduled every six hours, the election overhead effects amount to 0.004% of the network energy costs. The energy/bit results shown in Table 6-5 show that GMAC is able to produce a stable energy per data bit cost that is an order of magnitude less than those generated by SMAC and TMAC.

Table 6-5 WSN Network Energy/Bit vs. Network Packet Rate

	Packet Rate							
Protocol	60 pkt/s	40 pkt/s	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	2 pkt/s
SMAC	N/A	N/A	32.2μJ/bit	32.1μJ/bit	46.2μJ/bit	68.9μJ/bit	139μJ/bit	280μJ/bit
TMAC	35.3μJ/bit	36.3μJ/bit	40.1μJ/bit	42.0μJ/bit	45.3μJ/bit	52.2μJ/bit	73.0μJ/bit	116μJ/bit
GMAC	5.1μJ/bit	4.4μJ/bit	4.3μJ/bit	4.5μJ/bit	5.0μJ/bit	6.0μJ/bit	9.5μJ/bit	16.8μJ/bit
% GMAC Decrease	85.8%	87.8%	89.3%	89.2%	89.0%	88.4%	87.0%	85.5%

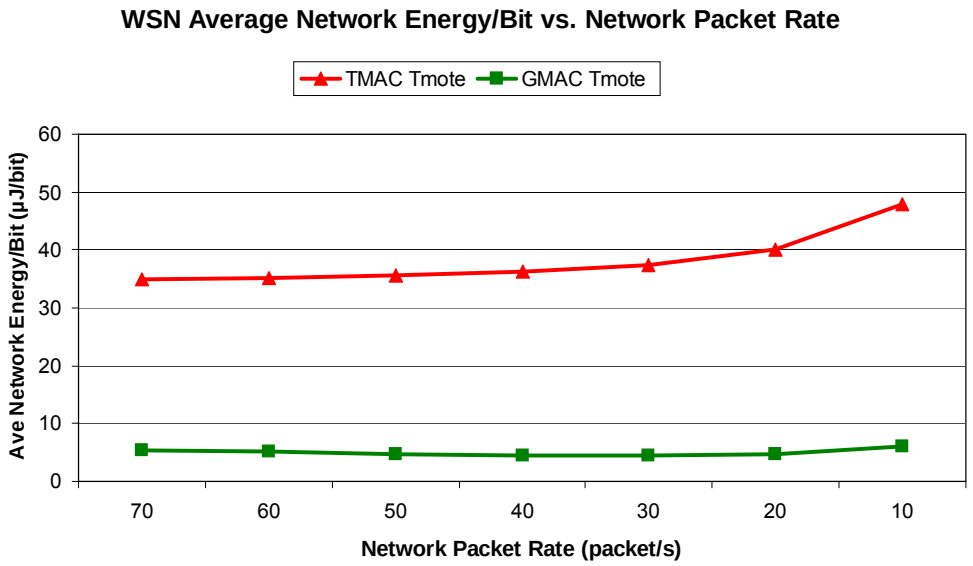

Figure 6-3 WSN Network Energy/Bit vs. Network Packet Rate

6.2.4 End-to-End Packet Delay vs. Unicast Traffic Packet Rate

One of the most significant costs in obtaining GMAC's increased network sleep percentage, increased network lifetime, and decreased network energy/bit is an increased data delivery delay. All regular GMAC nodes check their packet queues at the start of the collection period and make a one-time decision whether to wake up and contend for the medium. If a packet is queued immediately after that decision is made, the packet remains queued for another complete 500ms cycle + another collection cycle + (SIFS + GTIM) + partial distribution period. On average, a GMAC network packet will be queued halfway through a frame period, and wait 250ms until the next collection period, 250ms until the GTIM, 640us to 8 ms for a SIFS + GTIM, and 5 ms to 250ms for data distribution. The resulting range of 506 ms to 758 ms delivery time depends entirely on the packet rate. All of the GMAC end-to-end delays in Table 6-6 comply with this delay analysis. With TMAC and SMAC, packets queued for delivery during an active period are immediately able to contend for transmission. As shown in Table 6-6, TMAC end-to-end packet delay decreases as the active period increases with higher packet rates. Assuming that the average packet arrives in the middle of a frame, a packet would be able to immediately contend to send the packet after waiting 250 ms for the next active period.

Although channel contention time increases with increased packet rates, the window in which the network is active to send the packets also increases to effectively reduce the end-to-end latency. The SMAC nodes with packets generated during the active period can also immediately attempt to send them with a DIFS and contention backoff delay. The remaining portion of the frame, 1 – duty cycle, requires nodes to delay their traffic an average of (1-duty cycle) * frame time / 2. With the current simulation parameters, the average wait time for SMAC would be DIFS + average contention + (0s wait * 10% of frametime/2) + 90% of frametime/2 sleep wait = 0.56ms + 0 ms + 202.5 ms = 203 ms. Note that this number does not include the provision for increased backoff delay due to the packets accumulated during the inactive sleep period. All nodes attempt to send their accumulated packets upon simultaneous wakeup. Given the ultra-low 6 to 24 second per packet scenarios, the end-to-end packet delivery delays would be approximately 200 ms for SMAC and TMAC, and 500 ms for GMAC applications. Most of the WSN applications can tolerate the 750 ms maximum delay of GMAC and gain the significant increase in network lifetime.

Table 6-6 WSN Average End-to-End Packet Delay

End-to-End Packet Delay							
Protocol	60 pkt/s	40 pkt/s	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s
SMAC	N/A	N/A	N/A	395 ms	243 ms	223 ms	217 ms
TMAC	157 ms	185 ms	214 ms	219 ms	225 ms	231 ms	236 ms
GMAC	549 ms	537 ms	522 ms	519 ms	515 ms	512 ms	508 ms
% GMAC Increase	249%	191%	145%	137%	129%	122%	115%

6.2.5 Throughput

System throughput in these WSN networks is limited more by the impact on the network lifetime than the ability of this class of wireless sensor networks to respond to a burst of traffic. Both GMAC and TMAC have the ability to transfer up to 180 packets/sec with an average-sized IEEE 802.15.4 standard packet. Maintaining this packet rate would decrease the network lifetime to 5.8 days for TMAC and 19.2 days for GMAC. SMAC also has the ability to transfer

180 packets/sec if the active duty cycle was set to 100% prior to network deployment. The SMAC network lifetime is predetermined by the anticipated throughput requirement.

6.2.6 Network Lifetime vs. Broadcast Packet Rate

One of the techniques that GMAC employs in unicast message traffic to significantly outperform the energy efficiency of the other protocols is to eliminate the network-wide overhearing of all data-ACK exchanges. GMAC loses this advantage with broadcast traffic. Moreover, GMAC transmitting node must still request the reservation for the broadcast time slot. Figure 6-4 shows that the GMAC nodes require an additional FRTS and ACK for every broadcast data packet sent. The effects of this additional reservation exchange are not significant because only the transmitting node and gateway are awake during the collection period. Using the analytical model, the network lifetime results shown in Table 6-7 indicate that the advantages of the smaller GTIM overhead compared to the TMAC network-wide adaptive timeout are overcome by the added reservation exchange when the broadcast rate approaches 80 packets / sec. Again, this is an extremely high packet rate for networks designed for 6 to 24 second packet intervals.

Table 6-7 Broadcast Msg WSN Network Lifetime vs. Network Packet Rate

	Packet Rate							
Protocol	180pkt/s	140pkt/s	100pkt/s	80 pkt/s	60 pkt/s	40 pkt/s	20 pkt/s	1 pkt/s
SMAC	N/A	N/A	N/A	N/A	N/A	N/A	56.5 days	56.4 days
TMAC	8.9 days	11.3 days	15.4 days	18.9 days	24.4 days	34.5 days	58.6 days	174 days
GMAC	7.2 days	9.9 days	14.7 days	18.9 days	26.0 days	39.8 days	79.0 days	1001 days
% GMAC Increase	-18.8%	-12.6%	-4.9%	0.03%	6.3%	15.5%	34.8%	475%

	GMAC Broadcast Exchange				SMAC/TMAC Broadcast Exchange	
	FRTS-ACK→		GTIM-DATA		DATA – TA Timeout	
Transmitter	TX	RX	RX	TX	TX	RX
Gateway	RX	TX	RX	TX		
All others			RX	RX	RX	RX

Figure 6-4 GMAC and TMAC Broadcast Message Exchange Comparison

6.2.7 Network Lifetime vs. Number of Network Nodes

The energy efficiency of the GMAC protocol relies on the ability to distribute the gateway responsibilities among all of the cluster nodes to share the energy requirements and extend network lifetime. Table 6-8 and Figure 6-5 show that GMAC does not require a densely populated cluster to distribute the additional gateway energy consumption costs and save energy. As few as five nodes obtain more than a 100% increase in GMAC's lifetime over the TMAC network. This network lifetime increase grows to 786% with a 100 node network. Each of the scenario networks maintains a 4 packet/sec traffic load with the increasing number of nodes contributing to this rate. The slight dip in the parabolic curve is due to the x-axis change in increment from 5 nodes/division to 10 nodes/division. The TMAC and SMAC protocols are unable to leverage increased network energy capacity obtained with the increasing cluster size to gain network lifetime. GMAC's ability to schedule traffic and eliminate network-wide idle listening provides an immediate advantage for all cluster sizes. The network energy/bit consumption decreases rapidly with every additional node added to the GMAC cluster. The TMAC and SMAC protocols do not obtain any energy savings with the increase in nodes because all nodes incur the same idle listening and message overhearing energy costs.

Table 6-8 WSN Network Lifetime vs. Number of Network Nodes

	Number of Network Nodes						
Protocol	5 nodes	10 nodes	20 nodes	40 nodes	60 nodes	80 nodes	100 nodes
SMAC	56.7 days	56.6 days	56.6 days	56.6 days	56.5 days	56.5 days	56.5 days
TMAC	110 days	110 days	110 days	110 days	110 days	111 days	110 days
GMAC	240 days	392 days	585 days	778 days	876 days	935 days	974 days
% GMAC Increase	117%	257%	430%	608%	696%	746%	786%

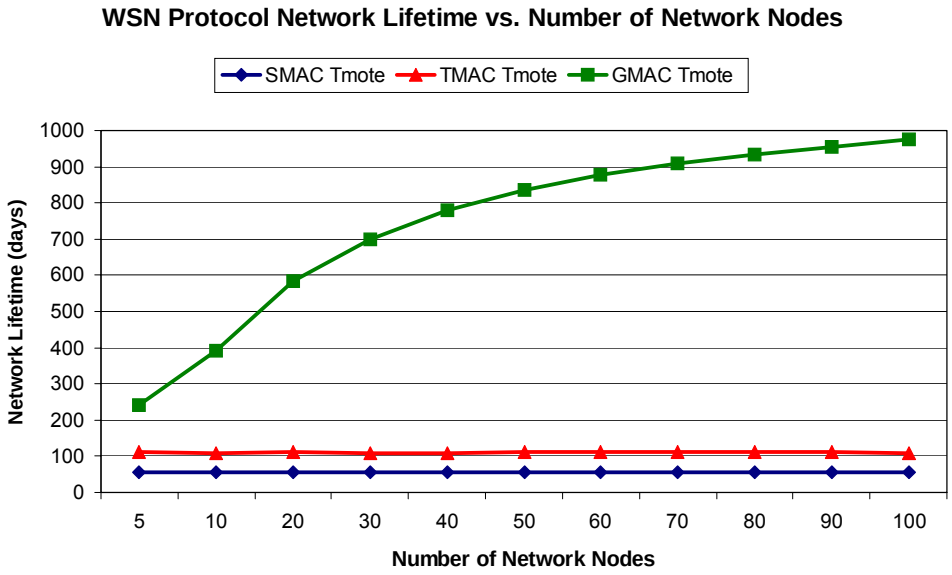

Figure 6-5 WSN Network Lifetime vs. Network Packet Rate

6.3 GMAC Comparison with 802.11 Power Save Modes

Two scenarios were designed to compare the effectiveness and efficiency of the GMAC protocol with the IEEE 802.11 DCF power save (PS) mode and the ad hoc PS mode as described in section 2.4.2. The representative protocol packet exchanges for GMAC and the 802.11 DCF PS modes are illustrated in Figures 6-6 and 6-7. A fair comparison with the 802.11 DCF protocol in PS mode is challenging because the 802.11 network requires an access point (AP) with robust memory and power resources. The two scenarios, one with a global power pool and one that discounts the AP's power consumption, fairly compare the three systems in each of their "intended" environments. Both scenarios employ the WSN packet sizes, protocol data rates, energy consumption models, and other parameters from the previously developed WSN network analytical model as a PHY/MAC protocol basis.

	802.11 AP Unicast Exchange						GMAC Unicast Exchange			
	RTS-CTS-DATA-ACK→ PS-POLL-DATA						FRTS-ACK→ DATA-ACK			
Transmitter	TX	RX	TX	RX			TX	RX	TX	RX
AP/Gateway	RX	TX	RX	TX	RX	TX	RX	RX		
Receiver					TX	RX			RX	TX

Figure 6-6 802.11 and GMAC Unicast Message Exchange Comparison

	802.11 AP Broadcast Exchange					GMAC Broadcast Exchange		
	RTS-CTS-DATA-ACK→DATA					FRTS-ACK→ DATA		
Transmitter	TX	RX	TX	RX	RX	TX	RX	TX
AP/Gateway	RX	TX	RX	TX	TX	RX	TX	RX
All others (N-2)					RX			RX

Figure 6-7 802.11 and GMAC Network Lifetime in WSN Ad Hoc Environment

6.3.1 WSN Ad Hoc Environment

The WSN ad hoc environment under study for typical GMAC device deployment is characterized by homogeneous energy- and memory-constrained nodes. GMAC rotates the gateway cluster-head duties among all of the nodes and effectively distributes the cluster-management energy and the packet storage memory costs. 802.11 PS normally uses a memory- and energy-rich AP to handle the majority of the traffic exchanges. Allowing the 802.11 AP and other stations to globally tap into a central "pool of energy" and determining network lifetime based upon the number of days taken for the entire network to expend all of the *pooled* joules provides a fair evaluation of how each of the three protocols uses the available network energy to manage the cluster/BSS/IBSS traffic. This comparison model does not compensate for the 802.11 AP memory advantage. The GMAC model does not allow the gateway to transition to sleep to provide a GMAC version which closer resembles the 802.11 PS model. When the gateway was allowed to transition to sleep, the GMAC model was significantly more energy efficient than the IEEE 802.11 AP model.

The IEEE 802.11 DCF PS protocol model was able to achieve a higher network lifetime than the GMAC (gateway awake) model for low-traffic unicast scenarios in the WSN ad hoc environment. Table 6-9 and Figure 6-8 show that the network lifetime crossover point for GMAC to outperform the 802.11 DCF PS protocol is 6 packets/sec. At the lower data rates, even though the 802.11 PS BTIM is 4 bytes larger than the GTIM, the ability for the associated 802.11 nodes to wake up and sample only every fourth BTIM beacon allows the network to conserve more energy. Above the 6 packets/sec rate, the energy costs associated with exchanging the data packet twice (up to the AP and down to the receiver) and the PS-Poll request exceeds the energy savings achieved from waking up every fourth BTIM. The IEEE 802.11 ad hoc PS model cannot compete with the other two protocols because it requires all nodes to remain awake throughout the entire ATIM window. Even though this analytical model allows 802.11 ad hoc protocol to dynamically set the ATIM window to the required size based upon the packet rate, the network-wide message overhearing costs significantly limit the network lifetime for all traffic conditions.

Table 6-9 WSN Ad Hoc Environment Network Lifetime vs. Unicast Network Traffic Rate

	Packet Rate					
Protocol	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	0 pkt/s
802.11 Ad Hoc PS	13 days	15 days	19 days	26 days	39 days	79 days
802.11 DCF PS	121 days	146 days	175 days	205 days	232 days	248 days
GMAC (Gateway Awake)	175 days	188 days	201 days	214 days	226 days	237 days
GMAC (Gateway Sleeps)	368 days	438 days	530 days	655 days	834 days	1026 days

The WSN ad hoc network broadcast analytical results listed in Appendix B show a similar comparison. The crossover point in which GMAC begins to outperform the 802.11 DCF PS model occurs at a higher 14 packets/s traffic rate because the receiving nodes do not have to send a PS Poll message like the unicast traffic case. Beyond 14 packets/s, the cost of transmitting the actual data twice makes the 802.11 DCF PS protocol less energy and bandwidth efficient than GMAC.

Overall, both the 802.11 DCF PS protocol and GMAC protocol provide similar network lifetime results in the scenario where the GMAC gateway node remains awake throughout the frame period to provide a fair comparison. The bottom row of Table 6-9 shows that the GMAC

scenario is able to achieve significantly better network lifetime when the gateway is permitted to transition to sleep. When taking the 802.11 AP memory requirements into account and the inability of the AP to distribute network energy by rotating the AP responsibilities, GMAC works better for homogeneous, self-configuring WSN networks.

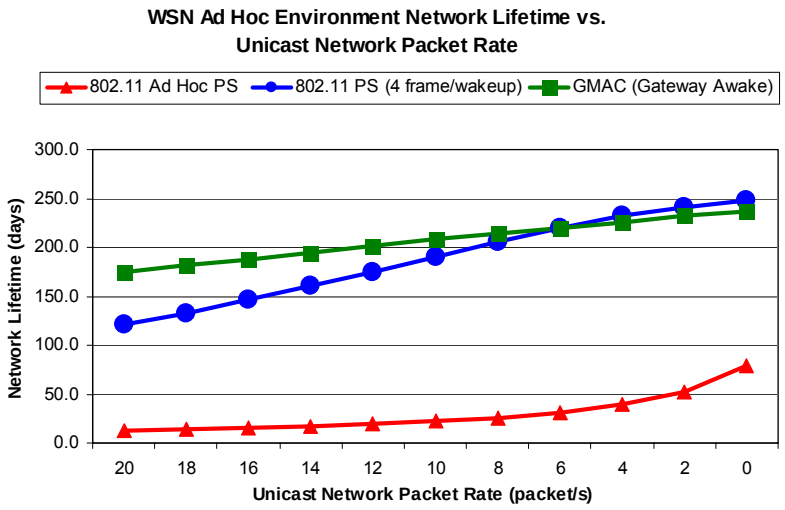

Figure 6-8 802.11 and GMAC Network Lifetime in WSN Ad Hoc Environment

6.3.2 WLAN PS Infrastructure Environment

The WLAN PS infrastructure environment is characterized by employing APs with robust energy and memory resources to provide energy-saving opportunities for the associated mobile, power-limited stations. Networks like the infrastructure-based IEEE 802.11 or the many two-tiered wireless sensor networks which have heterogeneous resource-rich nodes to perform the wireless distribution system functions require protocols which focus on expending the APs energy to preserve the energy of a resource-challenged, associated node. This new scenario excludes the AP/GMAC gateway power consumption to fairly evaluate the protocol's ability to extend the lifetime of a single resource-challenged node associated to a two-tiered, heterogeneous network. Since the scenario does not include the 802.11 AP's energy consumption, the comparison also does not include the GMAC gateway or *IBSS TBTT "winning" Node/temporary AP* overhead for fairness. This comparison method is inherently

biased in favor of the 802.11 infrastructure PS protocol because the GMAC gateway would normally sleep for a significant portion of the frame.

The IEEE 802.11 DCF PS protocol model was able to achieve higher network lifetime than the GMAC (awake) model for low-traffic unicast scenarios in the WLAN Infrastructure environment. Like the WSN ad hoc scenario in the previous subsection, Table 6-10 and Figure 6-9 show that the network lifetime crossover point for GMAC to outperform the 802.11 DCF PS protocol is 6 packets/sec. At the lower data rates, the ability for the associated 802.11 nodes to sample only every fourth BTIM allows the network to conserve more energy. Above the 6 packets/sec rate, the energy costs associated with the PS-Poll requests exceeds the energy savings achieved from waking up every fourth BTIM. Since the scenario is not including the energy consumption of the AP, the disadvantage of transmitting and receiving the data twice in the 802.11 DCF PS protocol is not a factor in determining associated node lifetime. As in the WSN ad hoc environment, the IEEE 802.11 Ad Hoc PS model cannot compete with the other two protocols because it requires all nodes to remain awake throughout the entire ATIM window.

Table 6-10 WLAN InfrastructureEnvironment Network Lifetime

	Packet Rate					
Protocol	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	0 pkt/s
802.11 Ad Hoc PS	23 days	27 days	33 days	41 days	54 days	109 days
802.11 DCF PS	209 days	296 days	443 days	705 days	1152 days	1694 days
GMAC (Gateway Awake)	443 days	537 days	659 days	819 days	1028 days	1293 days

6.4 GMAC Security: Broadcast Denial of Service Attack

Wireless sensor networks (WSN) offer the ability for applications to remotely monitor and react to events, but their remoteness also introduces challenges and vulnerabilities for network control and energy consumption. This section analyzes the security vulnerabilities from the denial of service attack described in Section 4.3.5 for GMAC and the other WSN energy-efficient MAC protocols. The main goal of a denial of sleep attack is to force a sensor platform to stay awake and receive a transmitted packet.

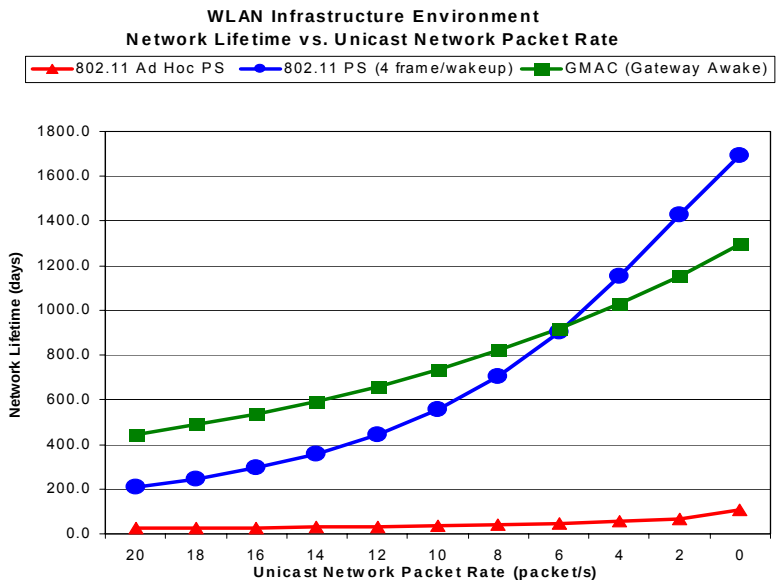

Figure 6-9 802.11 and GMAC Network Lifetime in WLAN Environment

SMAC's sleep cycle is fixed at the time of network deployment. This limitation causes the protocol to be inflexible in responding to network traffic fluctuations or network scaling. On the other hand, the fixed sleep cycle protects the network lifetime from the denial of sleep attacks by ensuring that nodes are only vulnerable during a fixed listening period. SMAC is most vulnerable to a broadcast attack in which an attacker sends multiple broadcast messages to keep all nodes active and receiving throughout the entire listening period. Let $T_{frame} = T_{listen} + T_{sleep}$ and $D = T_{listen}/T_{frame}$ (duty cycle $D \approx 10\%$). Since an attacker can broadcast a message to the entire network simultaneously, SMAC's maximum network lifetime from a broadcast attack is:

$$T_{network\,lifetime} = T_{sensor\,lifetime} = \frac{C_{battery(mAh)}}{(D)(I_{active(mA)}) + (1-D)(I_{sleep(mA)})} \tag{6-3}$$

SMAC is also vulnerable to unicast attacks with back-to-back RTS messages, but the attack effects are limited to draining energy from one node at a time, while denying all other nodes network access.

TMAC is more vulnerable to a broadcast attack than SMAC. If an attacker can get TMAC network nodes to receive repeated broadcast messages, the attacker can force all nodes to remain awake throughout the listen and sleep period, creating almost a 100% duty cycle. As shown in Eqn. 6-4, one attacker can drain the entire cluster's life.

$$T_{network\,lifetime} = T_{sensor\,lifetime} = \frac{C_{battery(mAh)}}{I_{active(mA)}} \tag{6-4}$$

Since an attacker can extend a broadcast message preamble across an entire sleep period and wake up every node, BMAC is equally as susceptible to a denial of sleep attack as TMAC.

GMAC has several energy-saving features which not only show promise in extending the network lifetime, but the centralized architecture also makes the network more resistant to denial of sleep attacks. Since cluster nodes only respond to the gateway node, network attackers cannot penetrate the link layer of the GMAC protocol. Unicast or broadcast messages sent to the gateway must be authenticated prior to being distributed to the individual nodes. The greatest success that a broadcast attacker can achieve is to send broadcast messages to the gateway and force the gateway to receive the entire message before discarding it due to authentication failure. Once the gateway's energy level reduces to a lower level, a fresh gateway takes its place. Shown in Eqn. 6-5, the attacker must effectively erode the network's energy one node at a time.

$$T_{network\,lifetime} = T_{sensor\,lifetime} = \frac{n_{nodes} * C_{battery(mAh)}}{I_{active(mA)}} \tag{6-5}$$

Any shared medium can be attacked with physical layer jamming. A denial of sleep attack is most effective if the attacking node uses knowledge of the MAC protocol to drain network energy without expending much of its own. For fairness, the model makes the unbiased assumption that each protocol can authenticate a message after it is completely received. The denial of sleep broadcast attack is modeled as an attacker sending four 75-byte broadcast messages per second. All nodes in the SMAC, TMAC, and BMAC protocols simultaneously receive a broadcast message, but discard the message if it fails authentication. The attacking node is able to deny sleep to all of the network nodes at the same time. As shown in Table 6-11, SMAC performs better than TMAC because it automatically transitions to sleep after receiving the last packet initiated prior to the sleep period. TMAC nodes must wait the additional adaptive timeout period. With GMAC, the attacker must gain access to the network through the gateway node which relays all inter-network traffic and reserves timing slots for intra-network traffic. If the gateway node does not properly authenticate the packet, it will not forward it to the other network nodes. The broadcast message does not directly affect the sleeping nodes. Furthermore, if the attacker cannot properly encrypt a GTIM message, the other nodes will not accept an attacker's traffic schedule if it tries to masquerade as the gateway node. Therefore, a link layer denial of sleep attacker can only affect one node at a time because nodes alternate the gateway responsibilities based upon incremental decreases in battery levels. Since n-1 nodes will always

be sleeping during the broadcast, the network lifetime for an attack increases linearly with the number of nodes. For these reasons, GMAC significantly outperforms the other MAC protocols in the broadcast denial of sleep attack category.

Table 6-11 Denial of Sleep Attack Performance Results

	Network Lifetime (days)	
MAC Protocol	Regular Broadcast Traffic	Broadcast Denial of Sleep Attack
802.11	6	6
SMAC	56	56
BMAC	83	83
TMAC	133	133
GMAC	443	938

6.5 Model Verification

As discussed in Section 3.8, model verification is the process of determining if a model was implemented correctly. Verification tasks include debugging the OPNET process code and ensuring that the system interfaces work as designed. The SMAC, TMAC, GMAC and RPM algorithm simulations were verified by using OPNET's integrated debugging tool. Additionally, the WSN OPNET models were verified by comparing for a tightness of fit with the analytical models.

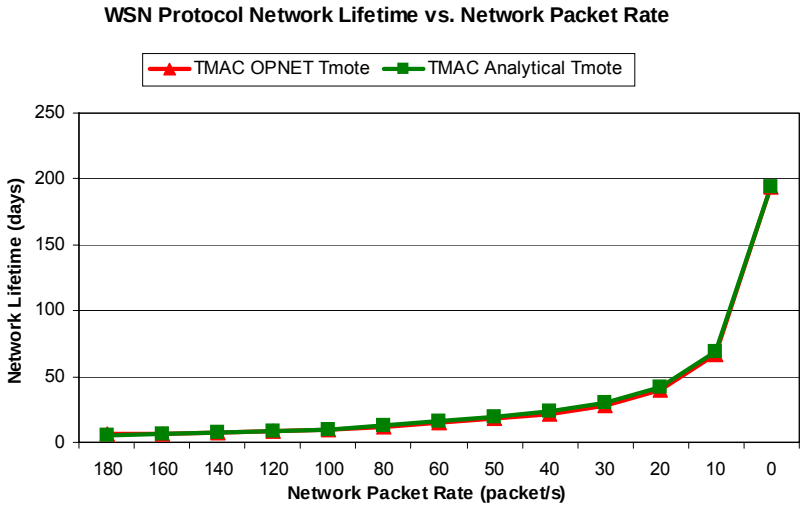

Figure 6-10 TMAC OPNET and Analytical Model Network Lifetime Comparison

SMAC achieved a network lifetime within 3% for all packet rates ranging from 0 to 20 packets/sec, which is SMAC's maximum packet rate with a 10% duty cycle. TMAC achieved a percentage difference of less than 7% for all packet rates up to 180 packet/sec. Figure 6-10 shows the two sets of TMAC network lifetime results. The small divergence is due to increased packet collisions in TMAC because of a more condensed packet window. Packet collisions are not incorporated into the analytical model. The GMAC simulation and analytical models were all within a 5% difference for packet rates ranging from 0 to 60 packets/sec, which is the maximum rate of the current OPNET implementation. Figure 6-11 shows the two sets of GMAC network lifetime results. GMAC should also be able to achieve packet rates of 180 packets/sec. Overall, the percentage difference between the analytical and simulation models averaged to 0.99% for SMAC, 4.38% for TMAC, and 2.64% for GMAC.

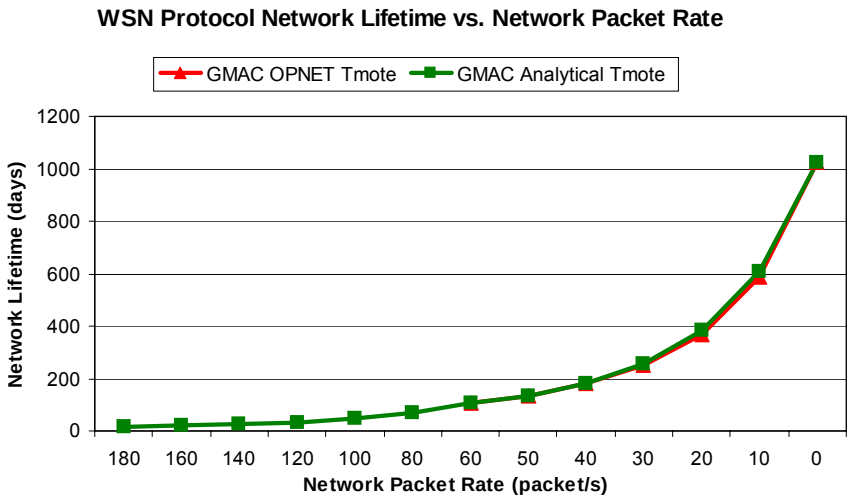

Figure 6-11 GMAC OPNET and Analytical Model Network Lifetime Comparison

6.6 Summary

This chapter presented the data and analysis for comparing GMAC with the other WSN and WLAN protocols using the network lifetime, sleep percentage, and end-to-end delay performance metrics. The OPNET Modeler simulation and MATLAB analytical models developed for GMAC and comparison WSN MAC protocols verify that GMAC's centralized

gateway approach achieved significant energy savings over the other WSN energy-efficient MAC protocols in both unicast and broadcast traffic models. GMAC's innovative WSN cluster management paradigm creates a traffic rhythm that extends node sleep durations and eliminates the primary source of network-wide energy loss, idle listening. The TDMA-based distribution period also eliminates network-wide message overhearing for all intra-cluster data exchanges. The performance results examined in this Chapter highlight GMAC's significant ability to extend network life. GMAC extended the ultra-low traffic network lifetime from 194 days to 1026 days (428% increase). In a unicast network scenario exchanging four packets per second, GMAC extended the network lifetime from 110 days to 834 days (657% increase). In comparing the energy per bit performance metric, GMAC achieved an increase of 87% energy savings as the leading protocol for unicast traffic. GMAC also outperformed the other WSN MAC protocols when sending broadcast traffic below a network rate of 80 packets/sec. In comparison to the IEEE 802.11 DCF protocol in power save mode, GMAC showed the capability of providing similar AP services while operating under severe memory and power resource constraints. An analysis of a denial of service broadcast attack showed that GMAC's centralized data reservation collection technique and dynamic energy-based cluster head rotation scheme successfully prevented an attacker from simultaneously draining every cluster node's energy as it did with the other protocols. In the GMAC protocol, the gateway node received and discarded the broadcast packet once it failed the message authentication code. With the other protocols, every node received and decoded the broadcast message before discarding it.

The next chapter presents the radio power management algorithm and the characterizations of the state-of-the-art WSN mote platforms.

Chapter 7

Radio Power Management (RPM) Algorithm

If everyone is thinking alike, then somebody isn't thinking.
--General George S. Patton

WSN designers strive to extend network lifetime while meeting application-specific throughput and latency requirements. Effective power management places sensor nodes into one of the available energy-saving modes based upon the sleep period duration and the current state of the radio. The newest generation of sensor platform radios with a 250 kbps data rate does not provide adequate time to completely power off the radio during overheard, 128-byte constrained, IEEE 802.15.4 transmissions. This chapter presents a new radio power management (RPM) algorithm which optimizes radio sleep capabilities by transitioning nodes to intermediate power level states. Additionally, the experimental work characterizes the radio power levels, state transition times, and state transition energy costs of two IEEE 802.15.4-compliant sensor platforms for improved accuracy in simulating WSN energy consumption. Overall, Chapter 7 focuses on developing the answer to one of the research questions posed in Section 1.4:

What criteria should be applied to optimize radio power management?

a. Energy costs at each power level

b. Transition energy costs between each level

c. Transition latency costs between each power level

7.1 Radio Power Management (RPM) Algorithm Introduction

The WSN radio power management (RPM) algorithm operating in the MAC layer sets the physical (PHY) layer radio low power modes (LPMs) based upon available sleep time. This RPM algorithm effectively regains short duration, power-saving opportunities lost with the newest generation of faster IEEE 802.15.4 low-rate wireless personal area network (LR-WPAN) -based [WPAN03] sensor platform transceivers. Short duration sleep offered by a network allocation vector (NAV) sleep mechanism provides significant energy savings [DaL03][YeH04][BrM06]. NAV sleep during message overhearing is significantly reduced because the new WSN platforms require more time to recover from sleep than is available during the shorter transmission time of the largest IEEE 802.15.4-compliant packet, 128 bytes. In addition to the RPM algorithm, the energy consumption model presented in this chapter provides increased simulation accuracy by incorporating the average radio energy consumption costs and transition times as the radio switches between transmit, receive, and LPM sleep levels.

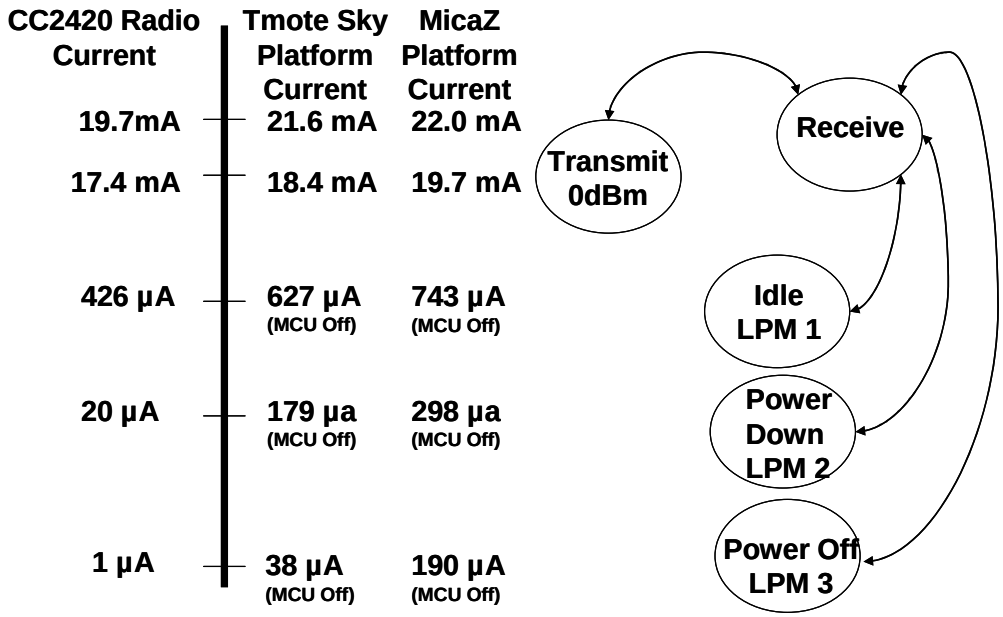

Figure 7-1 CC2420 Radio Energy Modes and Platform Energy Allocations

7.2 WSN Platform Energy Consumption Model

WSN network designers extend network lifetime by minimizing frame collisions, message overhearing, and idle listening. The most significant method in extending network lifetime is to synchronize nodes so that they actively pass data and then sleep as much as possible. Figure 7-1 shows that the CC2420 radio consumes up to 19.7 mA in the receive mode, but only 1 μA in power off mode. With typical 3000 mAh AA lithium batteries, the difference in lifetime of a fully active MICAz sensor mote platform (22.0 mA) and a sleeping platform (190 μA) is 5.7 days vs. 1.8 years (or battery shelf life).

Sleep transition measurements of the CC2420 radio integrated onto the MICAz platform indicate a 5.87 ms sleep and recovery transition time for the lowest LPM3 sleep mode. The average energy during the sleep transition is less than the receive mode, so time is the only transition cost. Effective power management places nodes into the various power-saving modes based upon the duration of the sleep period and can extend the lifetime of a network by two to three orders of magnitude. Previous communications platforms with effective data rates on the order of 46 kbps did not have a need for intermediate sleep levels. These low data rates provided nodes with sufficient time to completely power off and restart the radio during CTS-data-ACK transmissions. The new generation of radios with 250 kbps data rates transmits the data more rapidly and does not provide the time to completely power off the radio during overheard transmissions, but the nodes may be able to transition to an intermediate power-saving mode. Analyzing the 2.4 GHz, 250 kbps Chipcon CC2420 radio reveals three distinct power-saving levels: low power mode 1 (LPM1) through low power mode 3 (LPM3). LPM1 *idle* mode saves energy by turning off the radio frequency synthesizer which controls channel selection and up/down RF conversion. In addition to the frequency synthesizer, LPM2 *power down* mode also turns off the crystal oscillator which provides the timing reference for the entire radio chip. This step saves an additional 445 μA for the platform, but suspends all digital communications on the chip. The final radio power-saving level is the LPM3 *power off* mode. This mode turns off the voltage regulator which powers the radio chip. An interrupt from the microcontroller is required to restart the radio from this mode. LPM transition conditions require more than just the consideration of the available sleep time. Turning off the crystal oscillator in LPM2 with receive data waiting in the radio receive buffer would suspend the data transfer to the microcontroller.

The radio chip needs the timing signal generated by the crystal oscillator circuit to clock the data onto the system bus. The receive data would be delayed in LPM2, but not lost. Unfortunately, turning off the voltage regulator in LPM3 with data in either the receive or the transmit buffer would cause the data to be lost in the volatile radio RAM memory.

The *Platform Current* columns in Figure 7-1 show the static radio mode platform energy costs, and each state transition requires a transition time and average transition energy cost. Most WSN simulations do not adequately model the sleep transition costs. The simulation models either ignore the sleep transition energy costs or charge the transition to the highest energy state [LaH04]. The experimental current (I) measurements presented later in this chapter indicate that the average transition cost of 3.20 mA for a MICAz receive-LPM3-receive transition is an order of magnitude larger than the average LPM3 sleep mode base current (190 μA) and an order of magnitude smaller than the receive mode current (21.97 mA). Therefore, transition energy costs will have a impact on network lifetime. Additionally, the time required to recover from the LPM3 mode (5.87 ms) precludes many of the leading protocols from obtaining NAV sleep opportunities. Incorporating the RPM algorithm intermediate sleep modes allows these protocols to regain some of the energy savings. The platform characterization measurements taken for this research establish a power consumption model that increases the accuracy of WSN protocol simulation for future research and produces transition threshold parameters for the radio power management algorithm to optimize sleep transitions.

7.3 WSN Platform Characterization Experimental Circuit

Measuring micro-amp (μA) current consumptions and micro-second (μs) state transitions for simulation modeling and the RPM algorithm required developing an instrumentation circuit to amplify the signal prior to measurement on an oscilloscope. The platform current consumptions were determined indirectly by measuring the voltage across a special ohmic, low thermal noise resistor placed in series with the sensor platform's voltage source. The instrumentation circuit shown on the left side of Figure 7-2 details the connections to amplify the sensor platform's current flowing through the 1-ohm R_2 resistor [PoH04].. A fixed voltage source was used to provide consistent regulated power throughout the experiment. The right side of Figure 7-2 illustrates a block diagram of the instrumentation circuit and the method used to

calibrate the amplification circuit. To calibrate the circuit, a R_{mote} resistor was chosen to produce a typical mote 20mA current flow through the amplifier's input circuit across R_2. The voltage gain was established by comparing the instrumentation circuit output voltage to the input voltage using an Agilent 54622D Oscilloscope. Finally, the amplifier current gain was determined by accurately measuring the resistance value of R2 and applying Ohm's Law (I = V/R) to convert the circuit voltage gain to a 79.05 current gain. Analytical calculations presented in Appendix C confirmed the gain measurement within 0.7% of the theoretical current gain calculations.

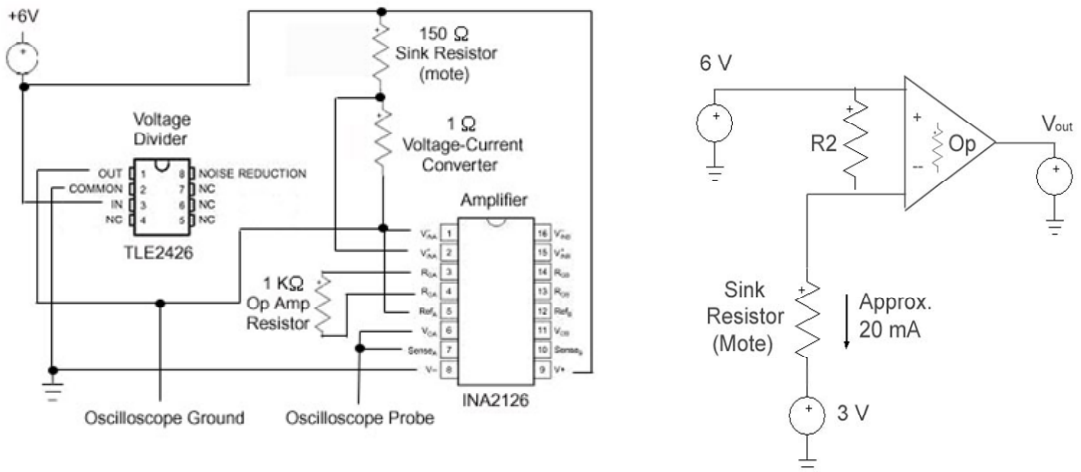

Figure 7-2 Instrumentation Circuit (left) and Equivalent Amplification Circuit (right)

7.4 Mote Platform Radio Low Power Mode Descriptions [NeB05]

The implementation and measurement of low power mode (LPM) radio transitions required a thorough analysis of the mote platform hardware and software components. The mote platform hardware was controlled by program files loaded into the platform microcontroller. A hierarchy of NesC-based software component modules packaged with the motes provided the commands for the TinyOS [TOS05] operating system to control the hardware. The CC2420 NesC radio control modules contained the function calls which interface the microcontroller with the radio. Platform general input/output (I/O) pins were used to analyze the transition function timing and provided feedback into each radio component initialization, start, and stop response times and energy costs. Programming the microcontroller to manage the LPM power levels

allowed the platform microcontroller, the Universal Asynchronous Receiver/Transmitter (UART) controller, and the radio to transition to the lowest required energy consumption state. Recovering from the LPM states required a succession of powering, resetting, and reconfiguring several subcomponents. The microcontroller recovered from the lowest LPM3 power down state by waking up using a watchdog interrupt and reestablishing the microcontroller-radio interface (Chipcon). Next, the microcontroller reset the voltage regulator (VReg) on the CC2420 radio. The crystal oscillator (Osc) was then reset and given time to stabilize to the proper frequency. Since powering down the CC2420 voltage regulator shut down the power to the radio's volatile RAM memory, the microcontroller must reload all of the radio parameters (radio address, radio frequencies, and optional operating modes). Next, the frequency synthesizer tuned the radio to the proper channel. The final step was setting the radio state to receive (Rx) and enabling two key interrupt handlers to indicate packet reception notification (FIFOP first in first out pin) and to communicate PHY preamble completion and MSDU message start and stop signals (SFD start frame delimiter). With all hardware subcomponents now enabled, the radio was now in the receive mode and prepared for incoming messages. The transition to sleep for each LPM level followed a reverse sequence of events as shown in Table 7-1. The oscilloscope screen captures in Figure 7-3 illustrate the various stages of the platform power up for the Tmote and MICAz platforms.

Table 7-1 LPM Hardware Transition Components

Turn On Mote	LPM1	LPM2	LPM3	Turn Off Mote	LPM1	LPM2	LPM3
Turn on Microcontroller (LPM3 only)	X	X	X	Disable Start Frame Delimiter Interrupt (SFD)	X	X	X
Reconnect Microcontroller-Radio Interface LPM3 only)	X	X	X	Disable Receive Detection Interrupt (FIFOP)	X	X	X
Turn on Voltage Regulator			X	Turn off Crystal Oscillator		X	X
Turn on Crystal Oscillator		X	X	Turn off Voltage Regulator			X
Reset Radio Registers			X	Disable Microcontroller-Radio Interface (LPM3 only)	X	X	X
Reset Radio Short Address			X	Turn off Microcontroller (LPM3 only)	X	X	X
Retune Radio	X	X	X				
Set Radio in Receive Mode	X	X	X				
Enable Receive Detection Interrupt (FIFOP)	X	X	X				
Enable Start Frame Delimiter Interrupt (SFD)	X	X	X				

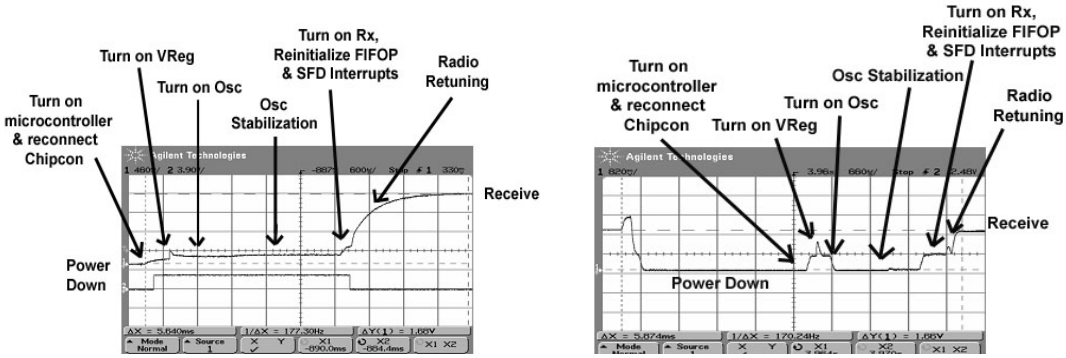

Figure 7-3 LPM 3 Transition Power Up Current Consumption: Tmote (left) and MICAz (right)

The experimental data for the energy costs and transition delays for each of the three LPM sleep levels are presented in Table 7-2. Each LPM transition was validated by periodically sending the motes to each of the LPM sleep levels and confirming the ability to send and receive messages upon recovery. The sleep and recovery transitions for each LPM sleep mode for the Tmote Sky and the Crossbow MICAz are shown graphically in the Figure 7-4 (a-c).

Table 7-2 Tmote and MICAz LPM Transition Responses

Low Power Mode	Total Transition Time (ms)		Average Transition Current (mA)		Average Base Current (mA)		System Effect
	TMote	MICAz	TMote	MICAz	TMote	MICAz	
Receive (RX)	-	-	-	-	21.56	21.97	
Transmit (TX)	-	-	-	-	18.40	19.70	
LPM1: Idle	4.56	4.38	3.72	3.04	0.627	0.743	Freq. Synthesizer Off
LPM2: Power Down	5.15	5.58	2.96	2.94	0.179	0.298	Crystal Oscillator Off Freq. Synthesizer Off
LPM3: Power Off	6.81	5.87	1.88	3.20	0.038	0.190	Voltage Regulator Off Crystal Oscillator Off Freq. Synthesizer Off

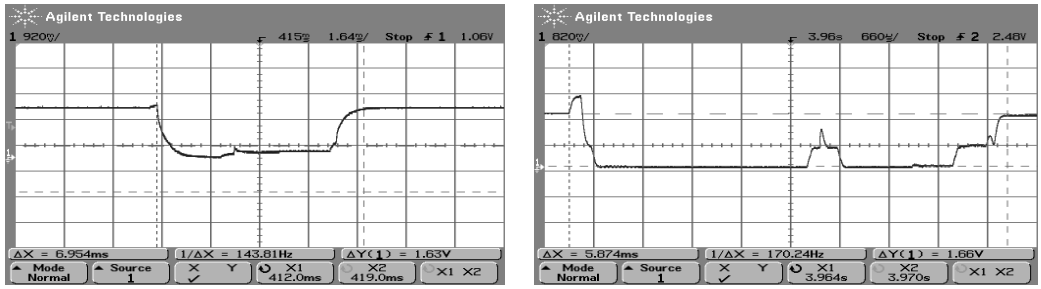

Figure 7-4a Tmote LPM3 (left) and MicaZ LPM3 (right) Current Levels.

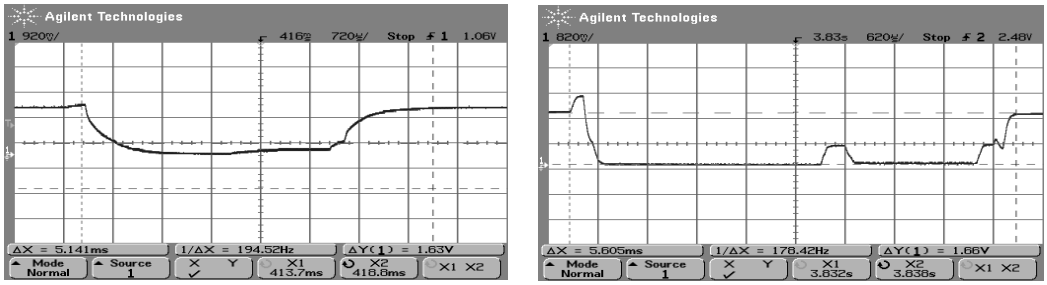

Fig. 7-4b Tmote LPM2 (left) and MicaZ LPM2 (right) Current Levels

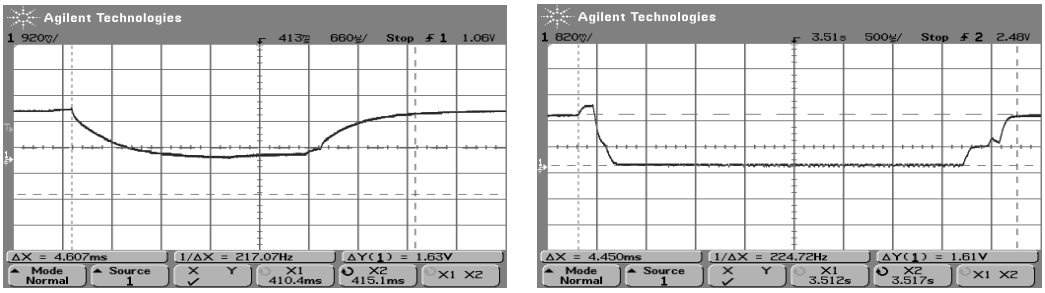

Fig. 7-4c Tmote LPM1 (left) and MicaZ LPM1 (right) Current Levels.

7.5 RPM Algorithm Design

The radio power management (RPM) algorithm creates graduated sleep modes for additional opportunities to transition the radio to lower power states. The Moteiv Tmote Sky [10] and Crossbow MICAz [11] WSN platform radio characterizations in this section offer experimentally-derived data to improve simulation accuracy and to optimize power-saving mode energy transitions for short duration sleep opportunities.

Integrating the radio power management (RPM) algorithm detailed in Figure 7-5 with WSN MAC protocols allows nodes to regain some of the short duration sleep opportunities lost with the faster 250 kbps IEEE 802.15.4 data rate. Previous technologies with slower data rates permitted nodes to transition to the lowest power mode (LPM3) for all data exchanges [YeH04]. If a node using the RPM algorithm is not the intended receiver of an RTS, the node uses the duration of the remaining CTS-data-ACK transmission sequence to optimize its power saving mode to LPM1, LPM2, or LPM3. While the experimentally-obtained CC2420 radio mode transition times in Table 7-2 establish the RPM transition thresholds, Table 7-3 illustrates the potential energy savings regained using LPM1 and LPM2 for the various packet data sizes and their associated RTS durations. The WSN hardware platforms only have the capability to support 128-byte packets. Since the WSN packets require an approximate 11-byte MAC service data unit (MSDU) header, the maximum data payload size in a data message is constrained to 117 bytes. Without RPM, nodes are only able to transition to LPM3. If NAV sleep is enabled, nodes attempt to transition to sleep for the short duration of an ongoing data transmission.

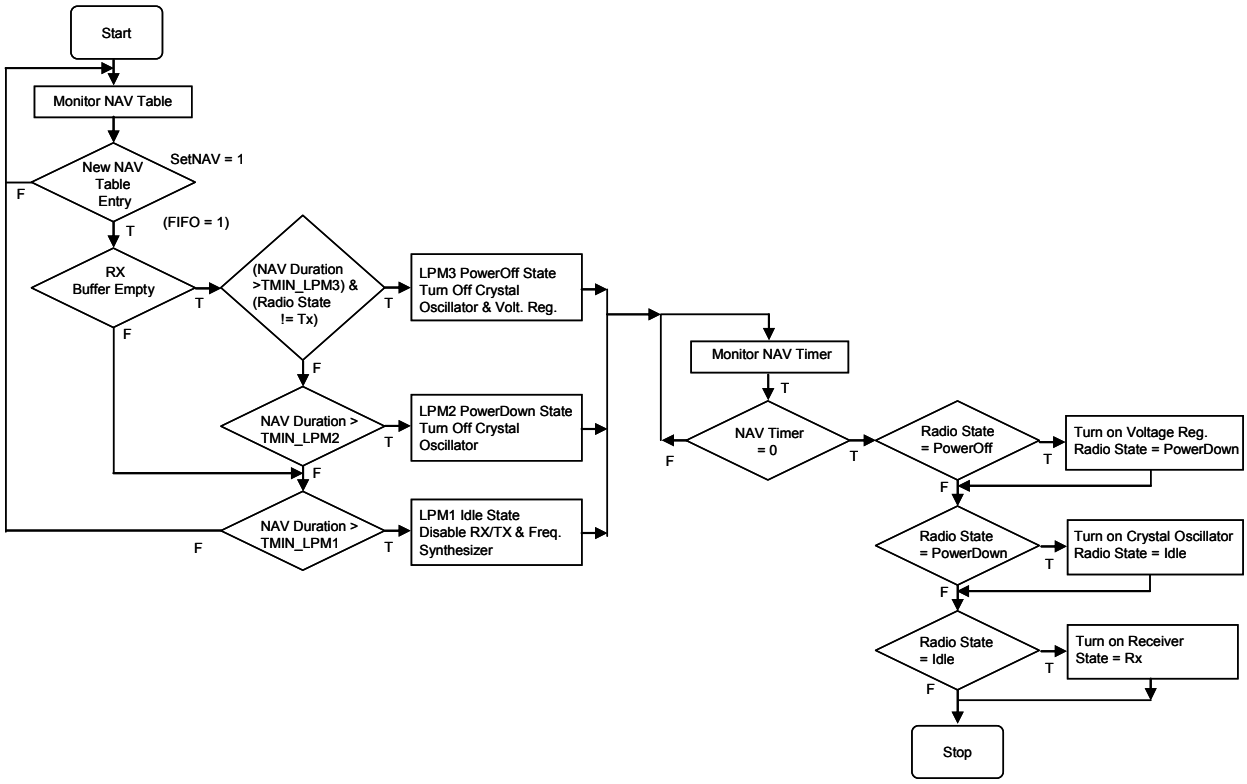

Figure 7-5 Radio Power Management Algorithm

7.6 OPNET RPM Simulation Model

A simulation scenario was designed in OPNET™ Modeler [OpM06] to compare the energy efficiency of the SMAC and TMAC protocols with message passing NAV sleep and the RPM algorithm. The scenario was composed of 20 WSN nodes operating with a 500 ms active/sleep frame period, and the SMAC protocol was set for a static 10% active duty cycle. The TMAC adaptive timeout algorithm produced a 13.48 ms idle channel sleep interrupt. The SMAC and TMAC models only permit LPM sleep transitions when the sleep duration request contains sufficient time to recover from an available sleep level. Additionally, the OPNET models charge the nodes for the transition energy costs (Table 7-2: transition time x average transition current x 3V battery voltage) and the appropriate LPM static base energy rate for the residual sleep duration. Although the sleep energy costs are lower than remaining in the receive mode and make any possible sleep transition an energy-saving event, these transition energy costs significantly decrease the expected network lifetime when compared to models which do not account for these transition costs. Finally, applying the 250 kbps data rate for the network represented the transmission speed of the new generation IEEE 802.15.4-compliant motes.

The scenario generated a uniform packet size distribution with a minimum outcome of 32 data bytes and a maximum outcome of 117 data bytes to represent an average WSN data exchange. The efficiency of NAV sleep was evaluated by simulating the SMAC and TMAC models both with message passing NAV sleep enabled and disabled. Next, the efficiency of the RPM algorithm was evaluated by simulating SMAC and TMAC with the RPM algorithm integrated into NAV sleep. Each model was simulated over a range of 0 to 20 packets/s to test energy efficiency over sparse to saturated traffic conditions. Although TMAC can extend its duty cycle to accommodate 180 data packets/s, the SMAC protocol's 10% active duty cycle limited the exponential packet generation rate to 20 data packets/s. The performance of the WSN models was then evaluated based upon network lifetime and average node sleep percentage. These performance metrics are defined as follows:

Network Lifetime is a measurement that can be categorized as either the time from network deployment to the first node failure or the time from deployment until the WSN

connectivity becomes partitioned. This measurement provides a fair evaluation of how all nodes work together as a system to extend network longevity. The SMAC and TMAC performance evaluations measure the time from network deployment until the failure of the first node. Network lifetime is expressed in days, and the performance rating increases with a higher number of days.

Sleep Percentage is a measurement of the amount of time nodes spend in any sleep state. Sleep percentage is calculated as the average time nodes spend in the LPM3, LPM2, or LPM1 sleep mode divided by the network lifetime. The performance rating increases with a higher sleep percentage.

Table 7-3 RPM Transitions based upon Packet Data Sizes

	IEEE 802.15.4 Data Packet Size (bytes)	
Radio Low Power Saving Mode	MICAz	Tmote Sky
LPM 1	81 to 105 data bytes	76 to 93 data bytes
LPM 2	106 to 114 data bytes	94 to 117 data bytes
LPM 3	115 to 117 bytes	None
Note: Packets limited to 117 bytes due to 11-byte MSDU Header		

A uniform packet size distribution scenario from 32 to 117 data byte packets provides insight into the effectiveness of message passing and the RPM algorithm. As shown in Table 7-3, the non-RPM assisted NAV sleep models which can only transition to LPM3 lose efficiency when the raw data size becomes smaller than 115 bytes for the MICAz because NAV sleep is limited to LPM3 mode. Likewise, the Tmote Sky is not able to transition to LPM3 sleep for the duration of a CTS-data-ACK exchange for any of the 117 data byte limited transmissions. In order for regular NAV sleep to be effective, data packet durations must be sufficiently long for nodes to enter the most energy-efficient LPM3 sleep level. Without sufficient time to transition to sleep, nodes remain awake in the receive mode for most packet exchanges. As illustrated in Figures 7-6 through 7-9, NAV sleep loses efficiency for both SMAC and TMAC in this simulation set due to the majority of the transmitted packets falling into the LPM2 and LPM1 sleep range. The RPM SMAC and RPM TMAC models outperformed the other non-RPM models by regaining the sleep lost by the faster IEEE 802.15.4 data rates and the slower recovery times.

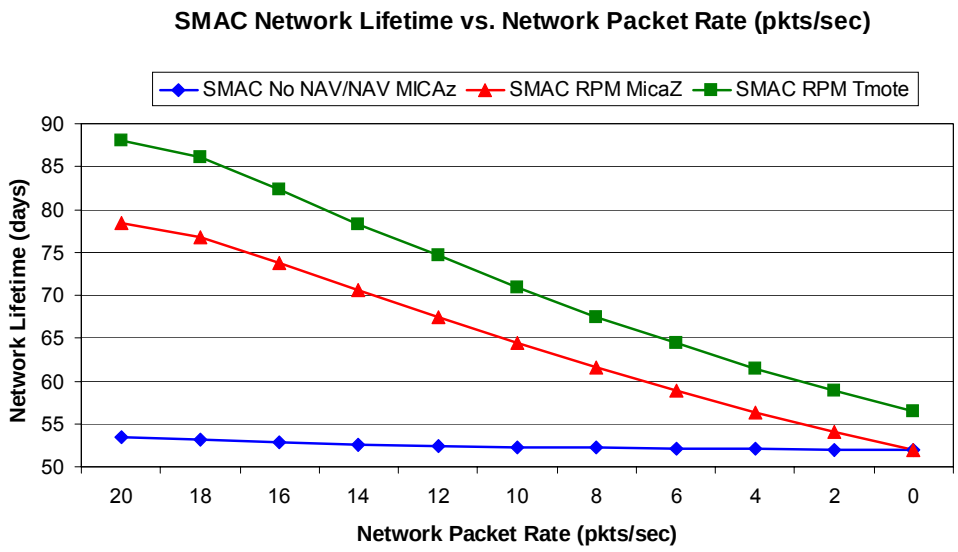

Figure 7-6 SMAC WSN Traffic Network Lifetime Performance

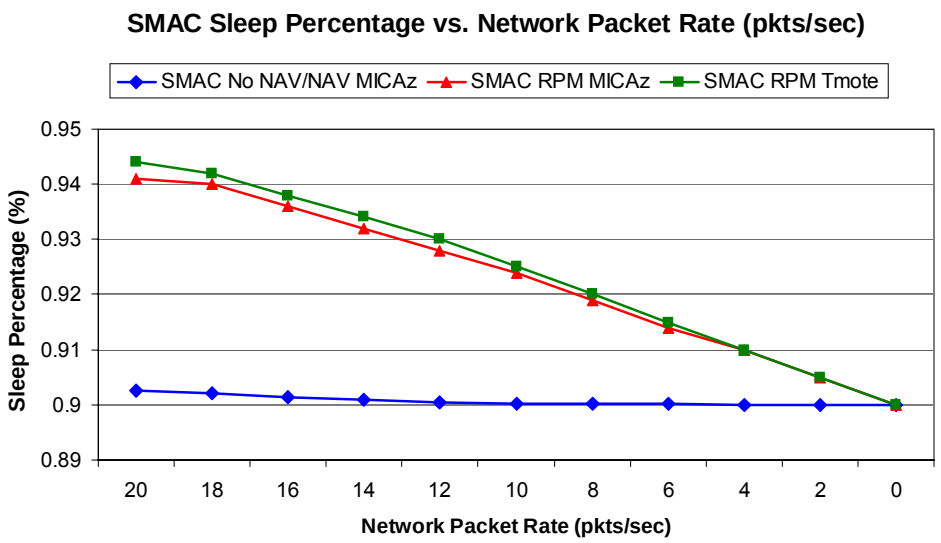

Figure 7-7 SMAC WSN Average Sleep Percentage

The SMAC model simulations in Figures 7-6 and 7-7 show that the message passing NAV sleep method was unable to save any appreciable energy over the No-NAV sleep model. The No-NAV sleep model was able to sleep only during the 90% inactive sleep cycle at a LPM3 level, and the NAV Sleep model was able to sleep only for the 90% static sleep cycle and the CTS-data-ACK duration for three packets sizes out of the range of 86 data packet sizes (32 data byte

packets to 117 data byte packets). By permitting the motes to sleep for the 90% static sleep cycle and the duration of the 37 LPM1, LPM2, and LPM3 packet sizes in the range 32 to 117 data bytes shown in Table 7-3, the MICAz RPM model was able to extend the network lifetime from 56.4 days to 78.4 days (32% increase). The Tmote Sky platform using the SMAC RPM algorithm extended the network lifetime from 56.4 days to 88.0 days (56% increase). Unless the network saturates, every frame cycle one of each TMAC node consumes a fixed 13.48 ms TA adaptive timeout listening cost in the receive mode. With no network traffic, TMAC networks are able to sleep 97.3% of the time and live 153.7 days with the MICAz. With the Tmote Sky, TMAC networks operate for 194.3 days under empty traffic conditions. Unlike the SMAC models, the NAV and RPM sleep during transmissions do not increase lifetime, these mechanisms only slow down the rate of network energy consumption. Figure 7-8 illustrates that the slope of the TMAC network lifetime vs. network packet interarrival time decreases with the RPM algorithm, extending the lifetime from 37.6 days to 51.7 days (37% increase) for the MICAz and 40 days to 56 days (40% increase) for the Tmote Sky. Compared with SMAC, TMAC had a lower network lifetime while operating at the SMAC saturation point because TMAC must remain in the receive mode for an additional 13.48 ms TA time beyond the SMAC 10% duty cycle.

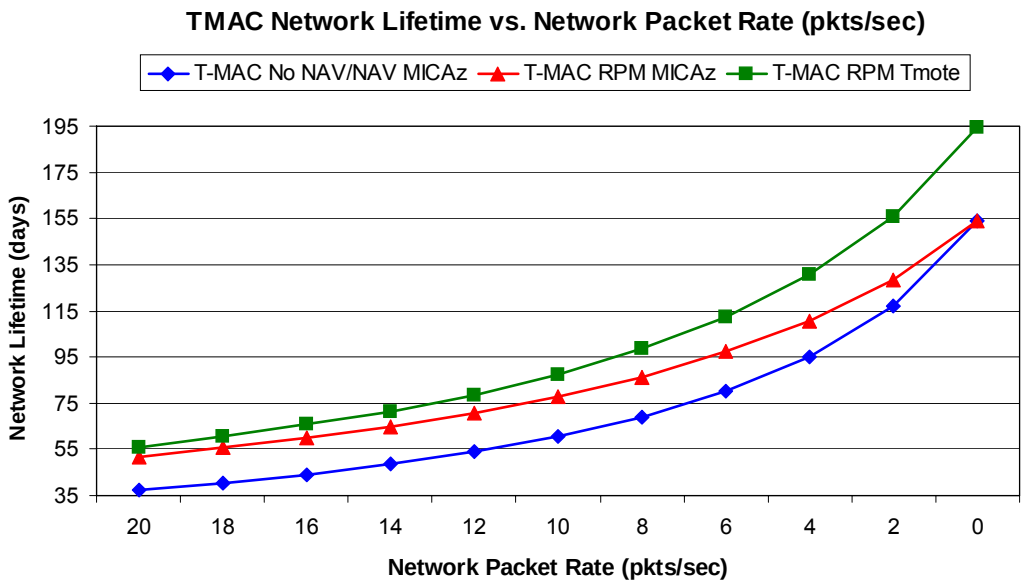

Figure 7-8 TMAC WSN Traffic Network Lifetime Performance

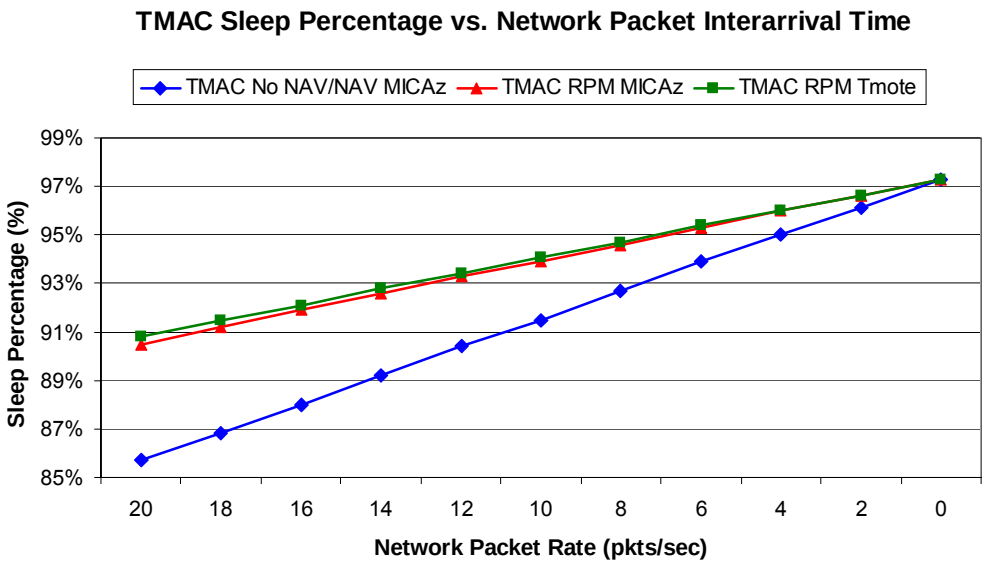

Figure 7-9 TMAC WSN Average Sleep Percentage

7.7 Summary

This chapter presented the data and analysis for characterizing the latest WSN mote sensor platforms and the design and implementation of the RPM algorithm. The experimental measurements characterizing the sleep mode transitions for the Moteiv Tmote Sky and Crossbow MICAz provide an accurate energy simulation model for future research and establish sleep transition thresholds for the proposed RPM algorithm. Also, the wireless sensor network radio power management algorithm exploits additional power-saving opportunities required for the newest generation of faster sensor platform transceivers. The RPM algorithm optimizes sleep transition decisions based upon the power and response characteristics of the sensor platform's transceiver. Implementing the RPM algorithm into a WSN MAC protocol demonstrated the ability to attain a 56% increase in the SMAC network lifetime and a 40% increase in the TMAC lifetime utilizing the current technology's realistic data patterns. The IEEE 802.15.4 WSN platform characterizations and the RPM algorithm provide the tools for researchers to continue their progress with the next generation of wireless sensor network platforms.

Chapter 8

Symbiotic Network

The great aim of education is not knowledge but action.

-- Herbert Spencer

8.1 Symbiotic Wireless Distribution System Network

Remotely deployed wireless sensor networks require innovative distribution systems to deliver the data back to the application monitoring program. This section introduces a novel approach to gathering data by offering a mobile user an incentive to collect, store, and forward data for an application, thus forming a symbiotic relationship. Overall, Chapter 8 focuses on developing the answers to two of the research questions posed in Section 1.4:

1. Can off-the-shelf IEEE 802.11b, g, and g+ wireless access points and network adapters provide adequate data exchanges between the roadside and mobile users traveling at highway speeds?

2. Can off-the-shelf IEEE 802.15.4 low-rate WPAN-based sensor platforms provide adequate data exchanges between the roadside and mobile users traveling at highway speeds?

8.1.1 Symbiotic Network Motivation

Data collection becomes more challenging with the ability to deploy sensors throughout the world. In a four-month, remote environmental habitat study, Szewczyk et al. discovered that the reliability of the backend infrastructure and the distribution network had a dominant impact on the overall network performance [SzM04]. Whether the application is biological sensors distributed across a city for Homeland Security monitoring or other sensors placed along a major highway to monitor traffic patterns, bridge structural strength, or weather conditions, gathering the data from low-power sensors normally results in a tradeoff between power and network lifetime. One solution to the problem is to create a symbiotic environment for network devices without power limitations to gather, store, and forward sensor messages from low-power sensors in order to gain network access from "hot-spot" wireless network access points.

Connecting highway vehicles to the Internet and collecting data from remote sensor networks are emerging fields which provide valuable services to the consumer, commercial, public safety, Homeland Security, and military markets. The symbiotic network integrates the static and mobile networks to provide the following mutual benefits: sensor networks gain data messaging to the Internet, and mobile stations gain Internet access in exchange for their willingness to forward the sensor data. For the traveler, highway Internet access provides web browsing access, email, route directions, roadway conditions, and local area services. In addition to capitalizing on the needs of the traveler, commercial applications also include electronic toll collection, fleet tracking, and on-board vehicular diagnostics reporting.

State Departments of Transportation (DOTs) could employ the Internet and supporting highway mobile ad hoc networks to gather sensor readings from remote sites or to transfer updates to electronic roadside information message signs. Fixed sensor sites report local weather conditions on the roadway (temperature, wind speed, road ice conditions, and visibility), bridge structural integrity, vehicular speeds, video, traffic congestion, and standard roadway usage statistics. DOTs also establish temporary networks to monitor traffic and program roadside hazard message signs throughout highway construction zones. If the sensors are near built-up

areas, many state DOTs currently contract dedicated communication lines to gather the information. For temporary construction zone applications, many states employ IEEE 802.11-based network bridging systems with multiple line-of-sight radios to link remote sites to leased communication collection points. This sensor data relay technique is both expensive and time-consuming to deploy. Likewise, the Department of Homeland Security (DHS) has a need to collect nuclear, biological, and chemical data readings from sensors all along the highway, both locally and remotely. The sensitivity of DHS data may require limiting the types of collection vehicles to trusted safety and law enforcement agencies. Finally, the military could use all aspects of the static sensor, mobile ad hoc, and infrastructure integrated system to transfer command, control, communications, computers and intelligence (C4I) data throughout the battlefield and theater of operations.

8.1.2 Symbiotic Network Architecture

The highway environment presents all of the major challenges faced in mobile ad hoc networks (MANETs). Some of the limiting features include reduced radio ranges, partitioned networks, lowered signal-to-noise ratios (SNR) due to the Doppler effect, and limited access to power in remote locations. The symbiotic network shown in Figure 8-1 mitigates many of these highway challenges by localizing all data transfers to short-range messaging between sensors, vehicles, and roadside access points. Short-range communication preserves the sensors limited energy resources, and ad hoc message passing permits all network users to access the Internet beyond their individual communications range.

The symbiotic network establishes a wireless distribution system (WDS) to transfer data from both the sensor field and MANET to the Internet. The sensors (S#) in the sensor field collect remote data and forward the information to the active gateway sensor (S1 → AGS2). The active gateway sensor then aggregates and transfers the data messages onto the highway to passing mobile ad hoc network stations (AGS2 → Car 1) to store and forward to the Internet through the roadside access points (Car 1 → Car 2 → Car 3 → Roadside AP → Internet). Each of the individual networks and interfaces is introduced and explained in this section.

Sensor networks offer the ability for applications to monitor and react to distant events, but their remoteness introduces challenges in network control and power. In an effort to make

inexpensive sensor platforms ubiquitous, these platforms have limited processing capability, memory capacity, and battery life.

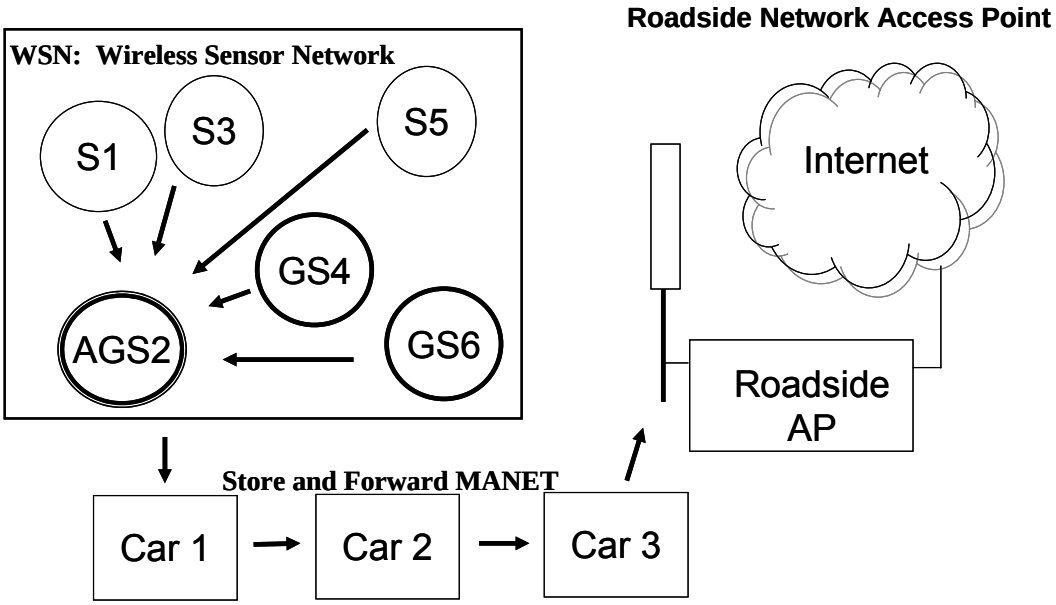

Figure 8-1 Symbiotic Network Architecture

In order to interface with the vehicular mobile ad hoc network without causing excessive energy drain on any one node, the gateway sensor nodes (GS#) in the symbiotic network rotate the data traffic and network responsibilities to share their resources in a manner that is self-adaptive to changes in topology, traffic loads, and existing battery conditions. The sensors in the sensor field may be homogeneous, but they must be able to directly communicate with the vehicles on the roadway to serve as a gateway node. Sensor networks attain long network lifetime by maximizing sleep periods while meeting the latency requirements of all of the sensors in the network [BrG05][PoH04]YeH04]. Gateway MAC (GMAC), specifically designed for the symbiotic network, collects sensor cluster data and forwards it to a mobile network in an energy-efficient manner. GMAC's innovative architecture is motivated by the requirement for wireless sensor networks to minimize the time radios spend in both the idle and receive modes in order to extend network lifetime. GMAC provides the ability for sensors to exchange data within a cluster for data fusion and forward messages out of the sensor network. Combining the advantages of both the contention- and reservation-based protocols, GMAC creates significant energy savings by employing a centralized node to gather all transmission requirements during a

contention-based period and then coordinating their distributions during a reservation-based, contention-free period. Then, without any additional overhead, the gateway duties are efficiently rotated among the nodes to spread out the increased network management energy requirements. After collecting all of the sensor field data messages, the gateway sensor node forwards the data to the mobile ad hoc network.

Next, the MANET collects the sensor data from the sensor gateway and forwards it toward roadside Internet gateway APs. To gather remote data in ad hoc networks, [SiB02] explored the use of remote nodes summoning dedicated data collectors using long-range radios and then exchanging data using short-range radios. The symbiotic network eliminates the need for a dedicated message collector and long-range radios due to the proximity of the sensor gateways to the abundant number of vehicles on the highway. Store-and-forward messaging bridges mobile network partitions which occur in sparse vehicular traffic conditions and enables reliable message traffic flow. Since highway vehicular traffic may stop during congestion, messages must be capable of propagating forward to the nearest access point. Highway experiments have shown vehicular ad hoc networks maintaining 1 Mbps communications using 802.11b devices within a 400m range of one another [ChK01]. Additionally, simulations have shown that the motion of vehicles on the highway in sparse vehicular traffic conditions decreased message delivery delay through store-and-forward routing [ZhA04].

Finally, the MANET delivers the sensor data to the Internet through roadside APs. FleetNet, an international consortium project initiated in 2000, began the development of linking together vehicles and connecting them to internet gateways along the road [FrE01]. Their objectives were to distribute locally relevant data and provide mobile users location-dependent information and services. The symbiotic network extends this concept by providing service incentives for the MANET stations to collect sensor data and encourages government agencies to build the infrastructure for the roadside Internet exchange.

Extensive research has been conducted to provide network routing in such a dynamic environment [BeF03] [Per02], and this experimental work evaluated the IEEE 802.11b, 802.11g, 802.11g+, and 802.15.4 link transfer capabilities at highway speeds.

8.2 Symbiotic Network Validation Testing Results

Experiments conducted on the Virginia Tech Transportation Institute (VTTI) Smart Road located in Blacksburg, VA, validated the symbiotic network capabilities for the IEEE 802.11 WLAN and IEEE 802.15.4 LR-WPAN networks to exchange data while traveling at highway speeds. Working with prototype CISCO IEEE 802.11g mesh-network equipment along the Smart Road afforded the opportunity to test a series of four access points (APs) in an extended service set (ESS) WLAN system joined by a wireless backbone routing system. Additionally, conducting experiments with a single AP with IEEE 802.11b, g, and g+ equipment provided insight to typical data rate capabilities for each mode at various mobile speeds. Finally, a roadside exchange between two IEEE 802.15.4 WSN mote platform systems at highway speeds proved the ability to effectively upload sensor data to passing cars.

8.2.1 Mobile Station to Single Roadside Access Point

The IEEE 802.11b, 802.11g, and 802.11g+ mobile station to single roadside AP performance evaluation experiments were conducted on the Virginia Tech Transportation Institute (VTTI) and the Virginia DOT Smart Road located in Blacksburg, VA. A pole-mounted, 12 dBi sector AP antenna and a 7.8 dBi omni-directional mobile antenna extended the mobile access radio coverage on the road. The performance of each of the protocols was evaluated on two individual quarter-mile segments of the road. Ixia IxChariot™ NetIQ endpoint tests measured throughput and delay for traffic directed from the mobile station to the AP at city and highway speeds – 20 mph through 70 mph. For consistency, each test employed the Buffalo Tech AirStation 125* high speed G54S access point with a wireless adapter using different compression and modulation settings for each of the 802.11b, g, and g+ experiments [Buf06].

The IEEE 802.11b/g/g+ roadside AP exchanges shown in Figure 8-2 illustrate that both 802.11b and 802.11g technologies provide sufficient throughput capabilities across the range of city and highway speeds. The throughput degradation for each protocol is primarily due to the reduced SNR caused by Doppler spreading. The 802.11b direct sequence spread spectrum (DSSS) encoded signal sustained 5 Mbps throughput over the entire range of speeds. Although the DSSS is less bandwidth efficient, the processing gain reduced the effective channel noise and increased energy per bit (E_b/N_o), thereby increasing the SNR. The 802.11g Orthogonal

Frequency Division Multiplexing (OFDM) modulated signal achieved a significantly higher throughput rate than 802.11b, but the performance degraded much more rapidly with both the distance from the AP and the vehicular speed. OFDM achieves a high spectral efficiency by dividing the available bandwidth into overlapping, orthogonal sub-carriers or sub-channels. The subsequent lower data rates of each of the sub-carriers reduce the multi-path distortion or delay spread, thus lowering inter-symbol interference (ISI) [Fla03]. These overlapping sub-channels cause OFDM to be particularly susceptible to inter-carrier interference (ICI) in the presence of the Doppler Effect. Delay spreading frequency shifts cause the sub-channels to no longer be orthogonal; therefore, their channels overlap and cause them to interfere with one another.

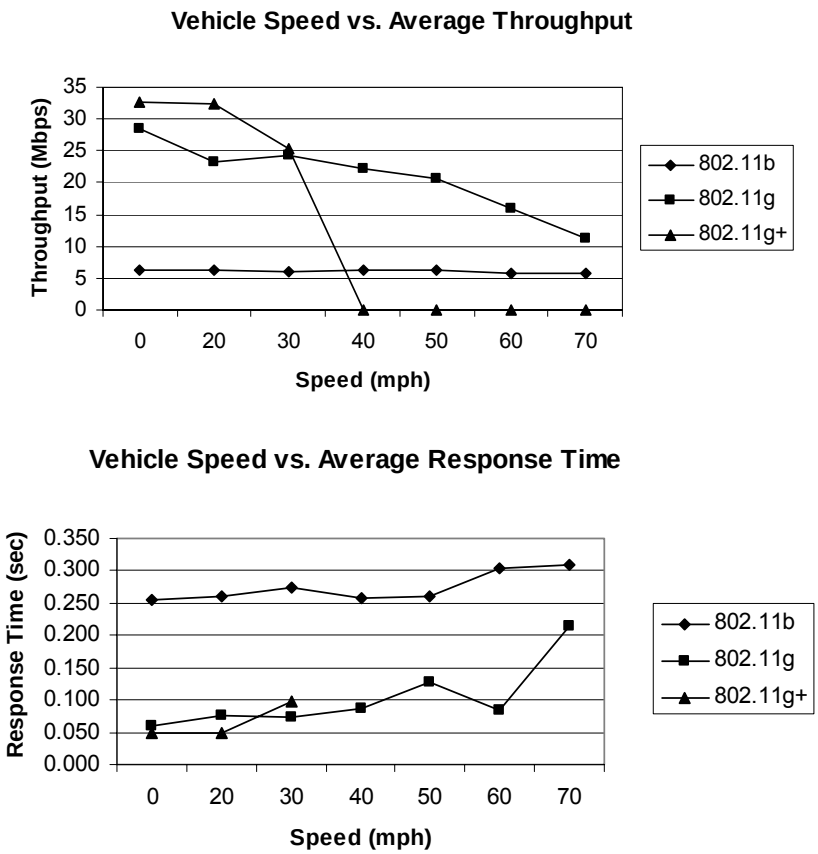

Figure 8-2 Mobile to AP Average Throughput and Response Time

The single AP roadside test also revealed that the advantages of the 802.11g+ high speed mode (125 Mbps physical rate) were only attainable at extremely low vehicle speeds. This technology typically employs dynamic packet bursting, fast frames, and hardware data

compression to achieve additional throughput. Dynamic packet bursting is an IEEE 802.11e quality of service (QOS) technique designed to increase throughput by decreasing the interframe spacing and successively sending frames without additional channel contention [KaM02]. While packet bursting increases the number of successively transmitted frames, fast frames increase the data payload to a negotiated size in order to increase effective throughput [AtC04]. The combined effects of these techniques reduce the SNR fade margin and allow the Doppler Effect to quickly attenuate the signal. This 802.11g+ failure occurred between 30 mph and 40 mph in the mobile station to single AP experiment.

8.2.2 Mobile Station to Multiple Roadside Access Points

An additional set of mobile to roadside AP tests evaluated the performance of linking overlapping APs with multiple radios and routers to form a wireless distribution system layout as shown in Figure 8-3. This configuration created a continuous extended service set (ESS) to deploy in highway locations not offering adequate line-of-sight coverage. Enhanced with pole-mounted, 13.5 dBi Yagii antennas for the WDS, a mobile test successfully evaluated IEEE 802.11g technology using CISCO routers and radios. Figure 8-4 shows the 20 mph and 60 mph results of the 802.11g ESS maintaining data rates between 10 Mbps and 20 Mbps as the vehicle traveled over a mile and accessed four APs. The data rate fluctuations were due to the time varying multipath signal fading which reduced the SNR and transmission data rate. The ESS performance can be optimized by tuning the AP handoff SNR and orienting the AP sector antennas to provide the maximum coverage based upon the anticipated handoff location. For example, in the 60 mph test shown in Figure 8-4, the mobile station maintained an association with AP2 beyond the point on the road where it passed AP3's sector antenna. In this case, tuning the SNR transition level to a higher threshold would cause the mobile device to switchover to AP3 earlier and increase the data throughput. The IEEE 802.11g ESS experimental results show that the network can maintain higher throughput levels than the maximum effective 5.5 Mbps associated with IEEE 802.11b technology at highway speeds.

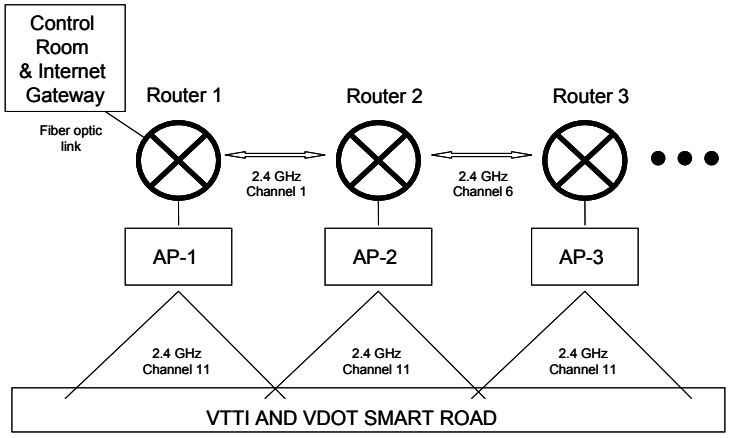

Figure 8-3 VTTI Smart Road WLAN Network

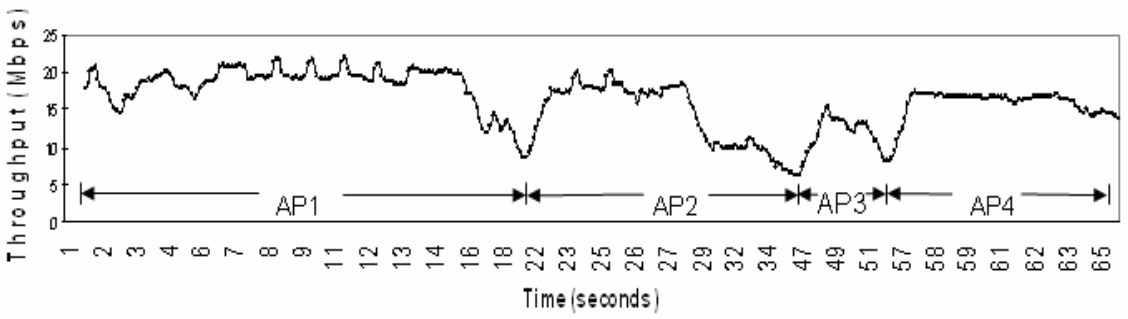

Figure 8-4 IEEE 802.11g 60mph Multi-hop Throughput

8.2.3 Gateway Sensor to Mobile Station

The gateway sensor to mobile station interface tests employed both Moteiv Telos A and Crossbow MICA2 low-power wireless sensor modules to verify the ability for IEEE 802.15.4 low-rate wireless personal area network (LR-WPAN) platforms to transfer data at highway speeds. The Telos 2.4 GHz transceiver modules utilized on-board, inverted-F microstrip antennas, and the Mica2 916 MHz transceiver modules employed an attached quarter-wave monopole antenna. Both platforms specified an approximate 125m outdoor transmission range. With the transmitting sensor modules placed on the roadside and elevated to one meter, a passing mobile module mounted on a car successfully received packets while traveling at speeds ranging from 20 mph to 70 mph.

Table 8-1 shows the experimental transfer capability results for each of the two platforms at various highway speeds. With a 30 byte packet size, the static baseline entry for each device reveals that the Telos CC2420 250 kbps radio is capable of transmitting a maximum 130 packets/s, and the Mica2 CC1000 76.8 kbps radio is capable of transmitting a maximum 32 packets/s. Also, a vehicle traveling 70 mph has the ability with the Telos A motes to collect more than 14,000 bytes in one pass. Each packet includes a 5 byte MAC header (2 byte address, 1 byte active message type, 1 byte group id, 1 byte payload length) and a 2 byte CRC field. Since every packet must contend for the channel and respond with a physical layer acknowledgement, the effective transfer rate will significantly increase as the size of the packet approaches the IEEE 802.15.4 128-byte maximum packet size. Another test showed that the 70 mph data transfer increased by 28% from 11,000 bytes to 14,100 bytes by increasing the Telos A packet size from 20 bytes to 30 bytes. With regard to vehicular velocity, analysis of the data showed that the Doppler Effect had almost no influence on the transfer of data at the various speeds for the DSSS 250 kbps transmissions. The reduced data transfer sizes for higher speeds were the result of reduced time within the transmission range of the static mote.

Speed Trial	Telos A Transfer	Mica2 Transfer
Static (baseline)	3,890 bytes/s	948 bytes/s
20 mph	49,400 bytes	6,050 bytes
30 mph	29,800 bytes	5,145 bytes
40 mph	24,900 bytes	3,920 bytes
50 mph	19,700 bytes	3,690 bytes
60 mph	15,400 bytes	2,660 bytes
70 mph	14,100 bytes	2,160 bytes

Table 8-1 Telos A and Mica2 Gateway Sensor to Mobile Station Transfer

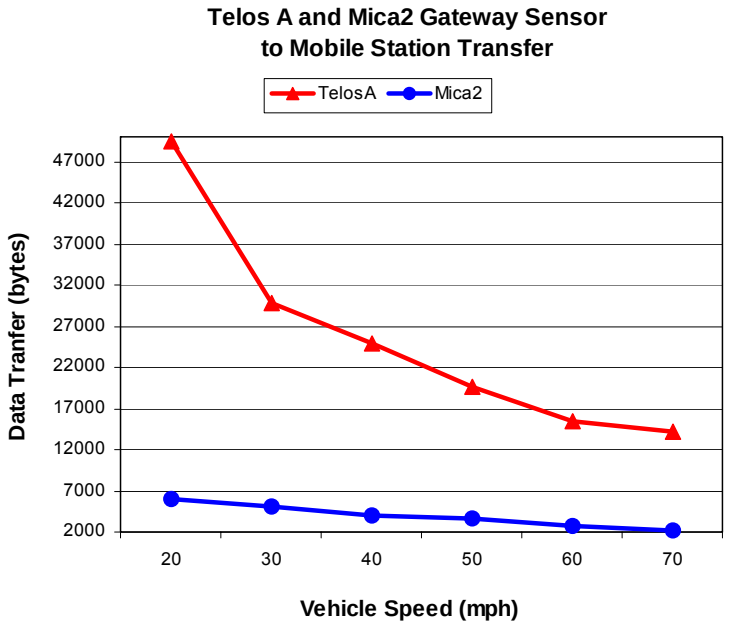

Figure 8-5 Gateway Sensor to Mobile Station Transfer

8.3 Summary

This chapter presented the background, data, and analysis of the VTTI and VDOT Smart Road experiments which validated that both the IEEE 802.11b/g WLAN mobile vehicle to roadside AP and IEEE 802.15.4 LR-WPAN roadside gateway sensor to mobile collector can effectively exchange data at highway speeds. The IEEE 802.11g mobile vehicle to roadside AP experiment showed that the technology supports sustained data exchanges with rates from 10Mbps to 15Mbps for highway speeds. The IEEE 802.15.4 data exchanges showed the capability of the new generation of wireless sensor platforms to transfer almost 50kB at 20mph and 15kB at 60mph. These experiments prove that the interface capabilities of the symbiotic network concept are a viable option as a wireless store-and-forward distribution system. The next chapter addresses future research and concluding thoughts.

Chapter 9

Conclusions

And when our work is done,
Our course on earth is run,
May it be said, 'Well Done;
Be Thou at Peace.'

--U.S. Military Academy Alma Mater
Paul S. Reinecke (USMA 1911)

This wireless sensor network research examined the development of an energy-efficient MAC protocol to extend network lifetime and monitoring capabilities. Designed for energy efficiency and robustness, GMAC dramatically improves on the network lifetime and energy/bit consumption costs. The OPNET Modeler simulation models for GMAC and the comparison protocols provide a means to conduct cost and performance tradeoff analysis. This chapter summarizes the work performed during this research dissertation and highlights significant achievements and extensions to the existing body of knowledge in wireless sensor networks.

9.1 Summary of Research

The primary goal of this research was to develop MAC layer energy-saving techniques to improve the network lifetime for remotely deployed wireless sensor networks. The GMAC

protocol developed throughout the research provides a robust, centralized coordination function which eliminates network-wide idle listening and significantly reduces energy consumption. The experimental measurements taken during this research characterizing the sleep mode transitions for the newest generation of WSN mote devices provide an accurate energy evaluation model for future research and establish sleep transition thresholds for the RPM algorithm. Finally, the development and validation of the Symbiotic network wireless store-and-forward distribution system offers a novel solution to the collection and distribution of sensor data from remote wireless sensor fields.

The chapters within this dissertation detail and document the research endeavors which will provide future researchers with a solid foundation to further develop energy-efficient wireless sensor network MAC protocols. Chapter 1 provided an introduction to the problem under investigation. The introduction established the purpose of enhancing the energy efficiency of these WSN MAC protocols and identified the key research questions to investigate and improve on the current techniques. Chapter 2 introduced the wireless sensor network and presented the challenges encountered in wireless medium communications. The energy-efficient techniques employed by the state-of-the-art WSN MAC protocols were integrated into the discussion of each of the primary sources of energy loss: idle listening, frame collision, protocol overhead, and message overhearing. Additional cluster management protocols described techniques to rotate the cluster head responsibility among the nodes, but these protocols failed to adequately consider the current energy state of each node in choosing the next cluster head. The chapter concluded with analysis of relevant design aspects of WSN MAC protocols: security challenges, timing requirements, platform hardware limitations, and application traffic considerations.

Chapter 3 elaborated on the research problem under investigation and provided specific objectives and an analytical research methodology to pursue, verify, and validate the results. The research employed Jain's ten-step performance evaluation method as a systematic approach to develop the GMAC protocol. Each performance metric was clearly defined, and verification and validation measures were established.

Chapter 4 introduced GMAC's energy-efficient design strategies and their improvements over the existing energy-efficient WSN MAC protocols for each of the four sources of energy loss. Additionally, a thorough discussion of GMAC's primary management functions provided

implementation details for the resource adaptive voluntary election (RAVE) scheme, the dynamic message exchange scheduling, the timing synchronization, and the security capabilities.

Chapter 5 presented the analytical and simulation models used to evaluate GMAC, TMAC, SMAC and IEEE 802.11 protocols using the network performance metrics defined in Chapter 3. These models establish the IEEE 802.15.4 PHY characterizations and MAC packet formats for future researchers to evaluate protocols. The energy models incorporate the optional RPM intermediate sleep levels and provide an extremely accurate mechanism to capture the energy costs of idle listening, contention, transmission, reception, and sleep in WSN MAC protocol communications. The analytical models provide rapid parameter tradeoff evaluation and help verify the complex OPNET simulation models.

Chapter 6 provided a comparison analysis of the analytical and OPNET Modeler simulation results of the GMAC, TMAC, SMAC and IEEE 802.11 (Power Save) protocols. Highlighting the effects of the performance metrics under the various traffic conditions showed that GMAC successfully outperformed the WSN MAC protocols using the network lifetime and energy/bit performance metrics. In comparison to the IEEE 802.11 DCF protocol in power save mode, GMAC showed the capability of providing similar AP services while operating under severe memory and power resource constraints.

Chapter 7 presented the RPM algorithm and the development of the Moteiv Tmote Sky and Crossbow MICAz mote platform characterizations. The radio characterizations obtained for the Tmote Sky sensor platform indicated that the new generation of transceivers with increased data rates and long duration sleep recovery times demanded a new multi-tiered power management approach. The experimentally-derived mote platform energy costs and transition times showed significant energy-saving sleep opportunities for existing WSN MAC protocols. Intermediate sleep states provided additional energy savings for short duration sleep.

Chapter 8 presented the Symbiotic wireless distribution system as an application for the GMAC protocol. The Symbiotic network represents a viable solution to gather data along a route of interest by offering the store-and-forward collection agent Internet access in exchange for the distribution service. Once the distribution infrastructure is in place, ad hoc networks can be seamlessly deployed to remote locations without planning costly reach-back communications. IEEE 802.11 and IEEE 802.15.4 experiments conducted on the VTTI Smart Road validated that

mobile traffic traveling at highway speeds can effectively collect and distribute remote sensor data to form a Symbiotic wireless distribution system.

9.2 Significant Contributions

This research makes four significant contributions to the state-of-the-art of wireless sensor networks. Most significantly, GMAC's centralized network management function offers substantial energy savings and lifetime extensions over existing wireless sensor network protocols. The GMAC protocol presents a new, coordinated WSN network paradigm which offers the ability to create a traffic rhythm that extends the duration of sleep opportunities and eliminates network-wide idle listening and message overhearing. The second major contribution is the introduction of a wireless sensor radio power management algorithm designed to exploit additional power-saving opportunities introduced with the newest generation of faster sensor platform transceivers. The total contributions to the WSN research field include:

1. Gateway MAC: a new energy-efficient wireless sensor network protocol.
2. RPM: a new radio power management algorithm.
3. Radio-mode power consumption and transition latency characterizations of the Moteiv Tmote Sky and the Crossbow MICAz wireless sensor platforms.
4. Validation of IEEE 802.11b/g/g+ and IEEE 802.15.4 data exchanges between a roadside and mobile platform traveling at city and highway speeds.

9.2.1 Energy-efficient WSN MAC Protocol

The WSN link layer MAC protocol presented in this research, Gateway MAC (GMAC), establishes a robust, centralized coordination function which eliminates network-wide idle listening and significantly reduces energy consumption. GMAC dynamically apportions TDMA slots according to the network traffic demands without imposing any network-wide message overhearing or idle listening overhead. GMAC increases the network lifetime by up to 400% for unicast traffic and decreases the energy/bit consumption by more than 80%. As shown in the simulation results, GMAC achieves significant energy savings in both heavy- and light-density

traffic environments by performing all required traffic scheduling operations while most of the network nodes are sleeping.

The resource adaptive voluntary election (RAVE) scheme developed for GMAC provides a passive, ad hoc cluster management election scheme that can extend beyond wireless sensor networks. This election scheme does not require passing control messages to determine the most eligible cluster head and provides a contention mechanism to randomly select a node from the most eligible group with minimal energy loss through collisions.

9.2.2 Radio Power Management Algorithm

WSN network designers extend network lifetime by minimizing frame collisions, message overhearing, and idle listening. The most significant method in extending network lifetime is to synchronize nodes so that they actively pass data and then sleep as much as possible. The newest generation of sensor platform radios with a 250 kbps data rate does not provide adequate time to completely power off the radio during overheard, 128-byte constrained IEEE 802.15.4 transmissions. These messages are transmitted too rapidly and do not provide the transition and recovery for the LPM3 sleep level. The radio power management (RPM) algorithm developed in this research optimizes radio sleep capabilities by transitioning nodes to shorter duration intermediate power level states. Incorporating the RPM algorithm's intermediate sleep modes has allowed two of the leading contention-based WSN MAC protocols to gain more than a 40% increase in energy savings, and the RPM algorithm can be immediately deployed into platforms utilizing the state-of-the-art, IEEE 802.15.4-compliant Chipcon CC2420 and CC1000 radio systems.

9.2.3 WSN Mote Platform Power and Latency Characterizations

The Moteiv Tmote Sky and Crossbow MICAz mote radio characterizations developed from research experiments supporting the RPM algorithm and WSN simulation models establish energy and latency transition costs for optimizing the radio power management (RPM) algorithm and enhancing the accuracy of evaluating future research simulation protocols. The sleep transition time and energy costs established the sleep transition thresholds for the RPM algorithm. Most wireless sensor network simulations do not adequately model the sleep transition costs. The simulation models either ignore the sleep transition energy costs or charge

the transition to the highest energy state. The Tmote Sky platform measurements reveal that the average transition cost of 1.88mA for a receive-LPM3-receive transition is two orders of magnitude larger than the LPM3 sleep mode current (37.6μA) and an order of magnitude smaller than the receive mode current (21.6mA). Additionally, the 6.81 ms transition and recovery time for the LPM3 mode precludes many of the leading protocols from obtaining the message overhearing sleep opportunities. Developing the power consumption and transition latency model increased the accuracy of WSN protocol simulation for future research and produced transition threshold parameters for the radio power management algorithm to optimize sleep transitions.

9.2.4 Symbiotic Network and Mobile AP interfaces

The validation tests of the IEEE 802.11b/g and 802.15.4 interface capabilities at highway speeds showed that the symbiotic network concept is a viable option for a wireless store-and-forward distribution system. The IEEE 802.11g equipment sustained more than a 15 Mbps throughput capacity in transferring data from a mobile station to an Internet gateway access point, and a pair of IEEE 802.15.4 wireless sensor platforms exchanged more than 15 kB of sensor data in one pass at 60 mph. The symbiotic sensor network provides a cost-effective solution for sensors to preserve their limited energy resources while transferring data from remote sites to locations around the world. Given the numerous commercial markets and the proven wireless technology, the symbiotic network is today's solution for merging the highway onto the Internet.

9.3 Future Research Directions

This WSN research presents MAC- and PHY-layer solutions to extending the network lifetime of wireless sensor networks. Significant future advancement of this work can be accomplished by implementing GMAC in hardware and deploying it to study and improve on its robustness. Next, complementing the GMAC protocol with a network layer such as AODV routing protocol and integrating the single-hop network into the multi-hop environment would provide insight into more comprehensive solutions for a deployable application.

Additional enhancements to the protocol can be gained by creating a message priority quality of service (QOS) system and optimizing the energy efficiency of the distribution schedule by placing receivers or transmitters with multiple messages into adjacent schedule slots.

9.4 Concluding Thoughts

This wireless sensor network research furthers the understanding of WSN MAC protocol energy-efficient techniques. The Gateway MAC (GMAC) protocol takes a step in a new direction by developing a WSN centralized cluster management system that excels in a deployed, ad hoc environment. Where other WSN protocols have created virtual clusters to reduce the amount of network-wide idle listening, GMAC achieves the elimination of network-wide idle listening and message overhearing without any additional protocol overhead. The GMAC collection and distribution strategy provides a scalable network for dense sensor fields, promotes fair data exchange, and utilizes the bandwidth efficiently. The dynamic allocation of the contention-free exchange time slots offers the same network scalability as contention-based schemes, but the contention-free period offers better network stability under heavy loads due to the scheduled nature.

The results of this dissertation have been published in five papers. The GMAC protocol was presented in the *IEEE Systems, Man, and Cybernetics (SMC) Information Assurance Workshop* [BrG05] and in the *IEEE Consumer Communications and Networking Conference (CCNC 2006)* [BrM06]. *The* GMAC protocol and the Symbiotic network were both presented in *IEEE Vehicular Technology Conference (VTC 2005)* [BrD05]. And the Radio Power Management algorithm was presented in both *OPNETWORK 2005* [BrF05] *and IEEE Wireless Communications and Networking Conference (WCNC 2006)* [BrF06].

As Moore's Law continues to drive the hardware capabilities and nanotechnology applications pervade the marketplace, the demand for remotely-deployed sensing applications will ignite the urgency for more innovative, energy-efficient networking solutions.

References

AtC04. Atheros Communications, "SuperG," Whitepaper, March 2004.

BaG02. L. Bao and J. Garcia-Luna-Aceves, "Hybrid channel access scheduling in ad hoc networks," In *10th IEEE International Conference on Network Protocols* (ICNP), pp 46-57, 12-15 November 2002, Paris, France.

BeF02. Bechler, M., Franz, W.J. and Wolf, L., "Mobile Internet access in FleetNet," In *Fachtagung Kommunikation in Verteilten Systemen* (KiVS 2003), Leipzig, February 2003.

BhD94. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless sensor LANs," In *Conf. on Communications Architectures, Protocols, and Applications*, pp 212-225, London, August 1994.

BrD05. M. Brownfield and N. Davis, "Symbiotic Highway Sensor Network," In *IEEE 62nd Vehicular Technology Conference (VTC)*, pp. 2701-2705, September 2005.

BrF05. M. Brownfield, A. Fayez, and N. Davis, "Wireless Sensor Network Radio Power Management," In *OPNETWORK 2005*, August 2005.

BrF06 M. Brownfield, A. Fayez, T. Nelson, and N. Davis, "Cross-layer Sensor Network Radio Power Management," In *IEEE Wireless Communications and Networking Conference (WCNC 2006)*, April 2006.

BrG05. M. Brownfield, Y. Gupta, and N. Davis, "Wireless Sensor Network Denial of Sleep Attack," In *6th Annual IEEE Systems, Man, and Cybernetics (SMC) Information Assurance Workshop (IAW)*, pp. 356–364, June 2005.

BrM06. M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis, "Wireless Sensor Network Energy-Adaptive MAC Protocol," In *IEEE Consumer Communications and Networking Conference 2006 (CCNC 2006),* Volume 2, pp. 778-782, January 2006.

Buf06. Buffalo Tech Corporation [Online], www.buffalotech.com, accessed February 2006.

ChK01. Z. Chen, H. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," In *ACM MobiHoc*, 2001.

Chi04. Chipcon Corporation, CC1000 and CC2420 low power and low voltage RF transceivers, http://www.chipcon.com/ , May 2004.

Cro06. CrossBow Corporation, MICA2 and MICAZ Series Sensor Processor / Radio Module. http://www.xbow.com , February 2006.

DaL03. T. van Dam and K. Langendoan, "Energy-efficient MAC: An adaptive energy-efficient MAC protocol for wireless sensor networks," In *Proceedings IEEE International Conference on Embedded Networked Sensor Systems (Sensys)*, pp.171- 180, Los Angeles, CA, November 5-7 2003.

ElD04. A. El-Hoiydi, J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks, In *AlgoSensors 2004*, Turku,, Finland, July 2004.

ElD03. A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux. Poster abstract: WiseMAC, an ultra low power MAC protocol for the WiseNET wireless sensor network. In *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, November 2003

ElG02. J. Elson, L. Girod, and D. Estrin, " Fine-Grained Network Time Synchronization using Reference Broadcasts," In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA. December 2002.

ElH02. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," In *IEEE International Conference on Communications (ICC)*, New York, April 2002.

EvR05. Ever Ready Battery Company. http://data.energizer.com/PDFs/l91.pdf, 2005.

FCC01. Federal Communications Commission, "Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz," Title 47, Vol. 1, U.S. Government Printing Office via GPO Access, October 2001.

Fla02. Flarion Technologies, "OFDM for mobile data communications," Whitepaper, March 2003.

FrE01. W. Franz, R. Eberhardt, T. Luckenbach, "FleetNet - Internet on the road," In *8th World Congree on Intelligent Transport Systems*, Sydney, September 2001.

GaK03. S. Ganeriwal, R. Kumar, M. B. Srivastava, "Timing-sync protocol for sensor networks," *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems* (SenSys 2003).

Gas02. M. Gast, 802.11 Wireless Networks: The Definitive Guide, O'Reilly, 2002.

GeK00. M. Gerla, T. Kwon, and G. Pei, "On demand routing in large ad hoc wireless networks with passive clustering," In *Proceedings of IEEE WCNC 2000*, Chicago, Illinois, September 2000.

HeC00. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," In *Proc. 33rd Hawaii Intl. Conf. on System Sciences*, January 2000.

HiC01 J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," Technical Report, U.C. Berkeley, 2001.

IETF04. Internet Engineering Task Force (IETF), "Ad hoc On-Demand Distance Vector (AODV) Routing draft-ietf-manet-aodv-13," Mobile Ad Hoc Networking Working Group, February 2003.

Jai91. Jain, R., The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, John Wiley and Sons, Inc., New York, 1991.

JuV02. E.-S. Jung and N. Vaidya, "An energy efficient MAC Protocol for LANs," In *Proceedings Joint IEEE Joint Conference of the IEEE Computer and Communications Societies (Infocom),* pp. 1756-1764, June 23-27, 2002.

KaK99. J. Kahn, R. Katz, and K. Pister, "Next century challenges: mobile networking for 'Smart Dust'," In *Proceedings of the International Conference on Mobile Computing and Networking*, pp. 271-278, August 15-20, 1999.

KaM02. S. Kandala, P. Mishra, F. Howley, et. al., "Status of Project IEEE 802.11e MAC Enhancements for Quality of Service," IEEE Working Document 802.11-02/604r1, September 2002.

KaS04. C. Karloff, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks, *ACM Embedded Network Sensor Systems (Sensys '04)*, pp. 162-175, November 2004.

Kar90. P. Karn, "MACA -a new channel access method for packet radio," In *9th ARRL Computing Networking Conference*, pages 134–140, September 1990.

KaE03. R. Karp, J. Elson, D. Estrin, and S. Shenker, "Optimal and Global Time Synchronization in Sensornets," CENS Technical Report 0012, April 10, 2003.

KuA04. S. Kulkarni and M. Arumugam, "TDMA service for sensor networks," In *24th Int. Conf. on Distributed Computing Systems (ICDCS04), ADSN workshop*, pages 604–609, Tokyo, Japan, March 2004.

KuR03. J. Kurose and K. Ross, Computer Networking: A top-down approach featuring the Internet, Addison-Wesley, Boston, Mass., 2002.

LaH04. K. Langendoen and G. Halkes, Energy Efficient Medium Access Control, Delft University of Technology, Netherlands, Technical Paper to appear in Embedded Systems Handbook, CRC.

LaH05. Y. Law, P. Hartel, J. den Hartog, and P. Havinga, "Link-layer Jamming Attacks on SMAC," Technical Paper, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, Enschede, The Netherlands, 2005.

LeM04. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gray, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for wireless sensor networks," In Ambient Intelligence., Springer-Verlag, 2004.

LiL04. J. Li and G. Lazarou, "A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks," In *3rd Int. Symp. on Information Processing in Sensor Networks (IPSN04)*, pp. 55–60, Berkeley, CA, April 2004.

Mac87. M.H. MacDougall, Simulating Computer Systems Techniques and Tools, MIT Press, Cambridge, MA 1987.

MaP02. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring. ACM International Workshop on Wireless Sensor Networks and Applications, Sep. 2002, pp. 88-97.

Mil94. D.L. Mills, "Internet time synchronization: The Network Time protocol," In Z Yang and T.A. Marsland, editors, Global States and Time in Distributed Systems. IEEE Computer Society Press, 1994.

Mul97. B. Mullins, "Cater: An Opportunistic Medium Access Control Protocol for Wireless Local Area Networks," PhD Dissertation, Virginia Polytechnic Institute and State University, June 1997.

Nit01. National Institute of Standards and Technology, "Federal Information Processing Standards Publication 197: Advanced Encryption System Standard (AES)," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, Nov. 2001.

Mot06. MoteIV Corporation, Telos Revision A Mote, http://www.moteiv.com, February 2006.

NeB05. T. Nelson and M. Brownfield, "Wireless Sensor Network Platform Resource Power Management," technical paper [unpublished], Virginia Tech, 2005.

Omn05. Omicell http://www.omnicel.com/techspecs.htm.

OpM06. OPNET Modeler software. Available: http//www.opnet.com/products/modeler/home.html., [Accessed: 22 January 2006].

PaS04. S. PalChaudhuri, A. Saha, and D. Johnson, "Adaptive clock synchronization in sensor networks," IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks, pp. 340-348, 2004.

PeC01. G. Pei and C. Chien, "Low power TDMA in large wireless sensor networks," In *Military Communications Conference (MILCOM 2001)*, vol. 1, pp. 347–351, Vienna, VA, October 2001.

Per02. C.E. Perkins, "IP mobility support for IPv4," RFC 3220, IETF, January 2002.

PeS04. A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *ACM Communications*, vol. 47, pp 53-57, 2004.

PoH04. J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," In Proceedings IEEE International Conference on Embedded Networked Sensor Systems (Sensys), pp. 95-107, November 2004.

PoK00. G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no.5, pp 51-58, May 2000.

Ran04. S. Ransbottom, "Mobile Wireless System Interworking of 3G and Packet Aggregation for Wireless LAN," PhD Dissertation, Virginia Polytechnic Institute and State University, April 2004.

RaO03. V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-efficient MAC: energy-efficient collision-free medium access control for wireless sensor networks," in Proceedings IEEE International Conference on Embedded Networked Sensor Systems (Sensys), pp. 181-192, November 5-7 2003, Los Angeles, CA.

RFC3626 "RFC 3626 - Optimized Link State Routing Protocol (OLSR)," Networking Working Group, Project Hipercom, INRIA, October 2003.

RFM05. RF Monolithics, TR1001 868.35 MHz Hybrid Transceiver, http://www.rfm.com, April 2005.

SiB02. J. Singh, N. Bambos, B. Srinivasan, and D. Clawin, "Wireless LAN Performance Under Varied Stress Condition in Vehicular Traffic Scenarios," *IEEE Vehicular Technology Conf.*, 2002.

SiR98. S. Sing and C. Raghavendra, "PAMAS: Power Aware multi-access protocol with signaling for ad hoc networks," *ACM Comput. Communications Review*, vol. 28, no. 3, pp.5-26, July 1998.

SoG00 K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network,: in IEEE Personal Commincations, Vol. 7, Issue 5, pp. 16-27, October 2000.

Sta02. W. Stallings, "Wireless Communications and Networking", 1st Ed., Prentice Hall, 2002

StA00. F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," In *Proceedings of The 7th International Workshop on Security Protocols*, vol. 1796 of LNCS, pages 172-194. Springer-Verlag, 2000.

StK97. M. Stemm and R. Katz, "Measuring and reducing energy consumption of network interfaces in handheld devices," In IEICE Transactions on Communications, vol. E80-B no. 8, pp. 1125-1131, August 1997.

SzM04 R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler, "An analysis of a largescale habitat monitoring application," In *Proceedings of the Second ACM Conference on Embedded Networked Systems (SenSys)*, November 2004.

SzP04 R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, January 2004.

TeI04. Texas Instruments, MSP430F149 Ultralow-power Mixed Signal Microcontroller, http://focus.ti.com/lit/ds/symlink/msp430f149.pdf, June 2004.

TOS05. TinyOS, http://webs.cs.berkeley.edu/tos, 2005.

WiFi05. Wi-Fi Alliance, http://www.wi-fi.org/, 2005.

WLAN97. LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN Medium access control (MAC) and physical layer (PHY) specification, IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, 1997.

WPAN03. LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN Medium access control (MAC) and physical layer (PHY) specifications for Low-rate Wireless Personal Area Networks (LR-WPANs), IEEE, New York, NY, USA, IEEE Std 802.15.4-2003 edition, 2003.

YeH02. W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," In *Proceedings Joint IEEE Joint Conference of the IEEE Computer and Communications Societies (Infocom),* pp. 1167-1576, June 23-27, 2002.

YeH04. W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," in *Proceedings IEEE/ACM Networking Transactions*,Volume: 12, Issue: 3, pp. 493-506, June 2004.

ZaA04. W. Zhao, M. Ammar, and E. Zegura. "A Message ferrying approach for data delivery in sparse mobile ad hoc networks," In *ACM MobiHoc*, 2004.

ZeR05. T. Zheng, S. Radhakrishnan, and V. Sarangan. "PMAC: An adaptive energy-efficient MAC protocol for Wireless Sensor Networks," In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 65-72, April 2005.

Appendix A: GMAC State Diagrams

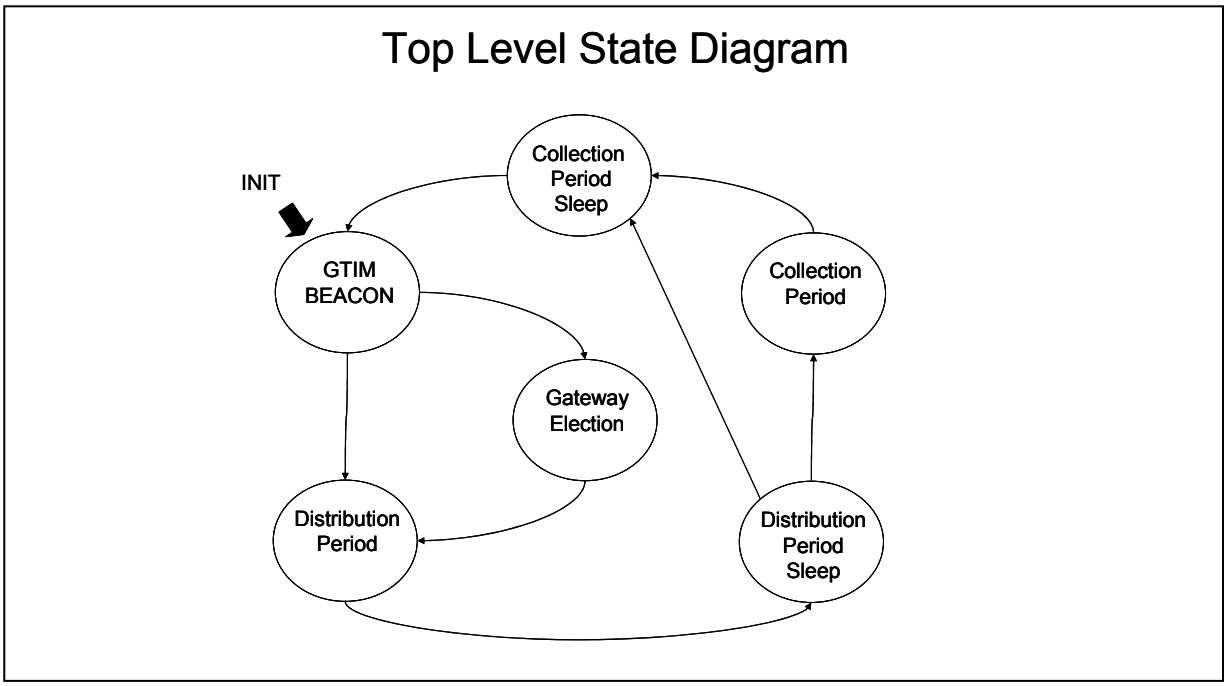

Figure A-1 GMAC Top Level State Diagram

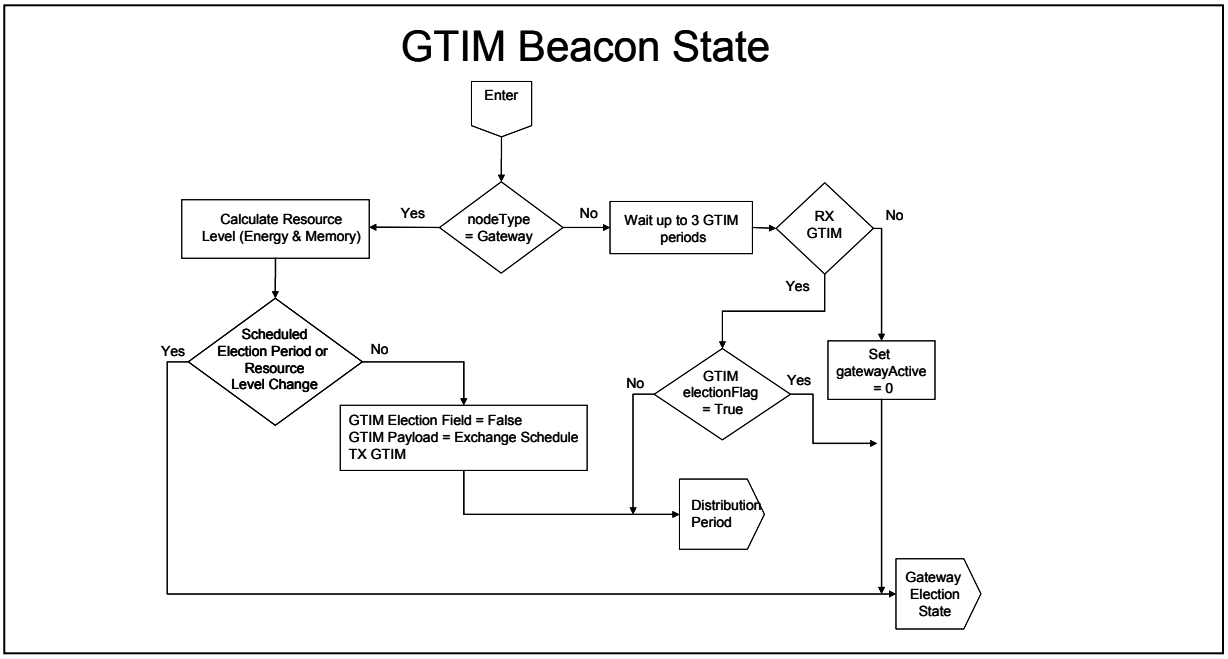

Figure A-2 GMAC GTIM Beacon State Flow Diagram

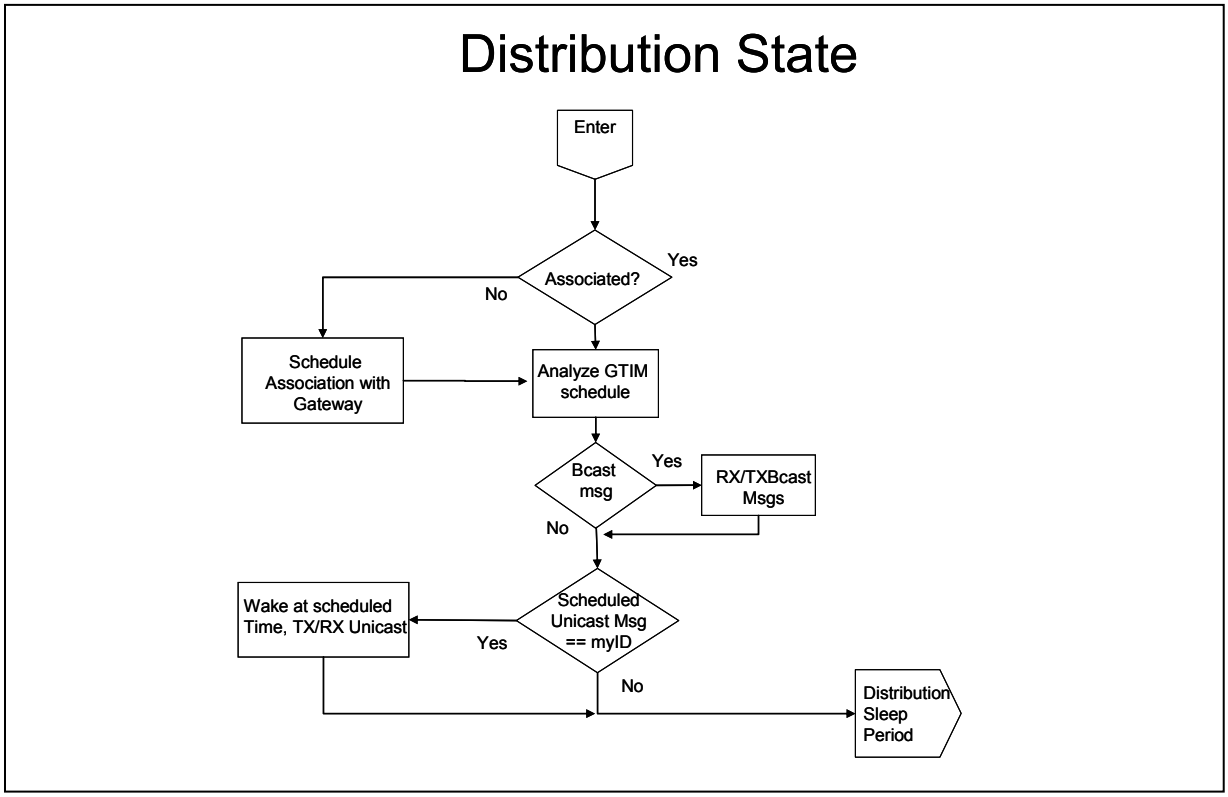

Figure A-3 GMAC Distribution State Flow Diagram

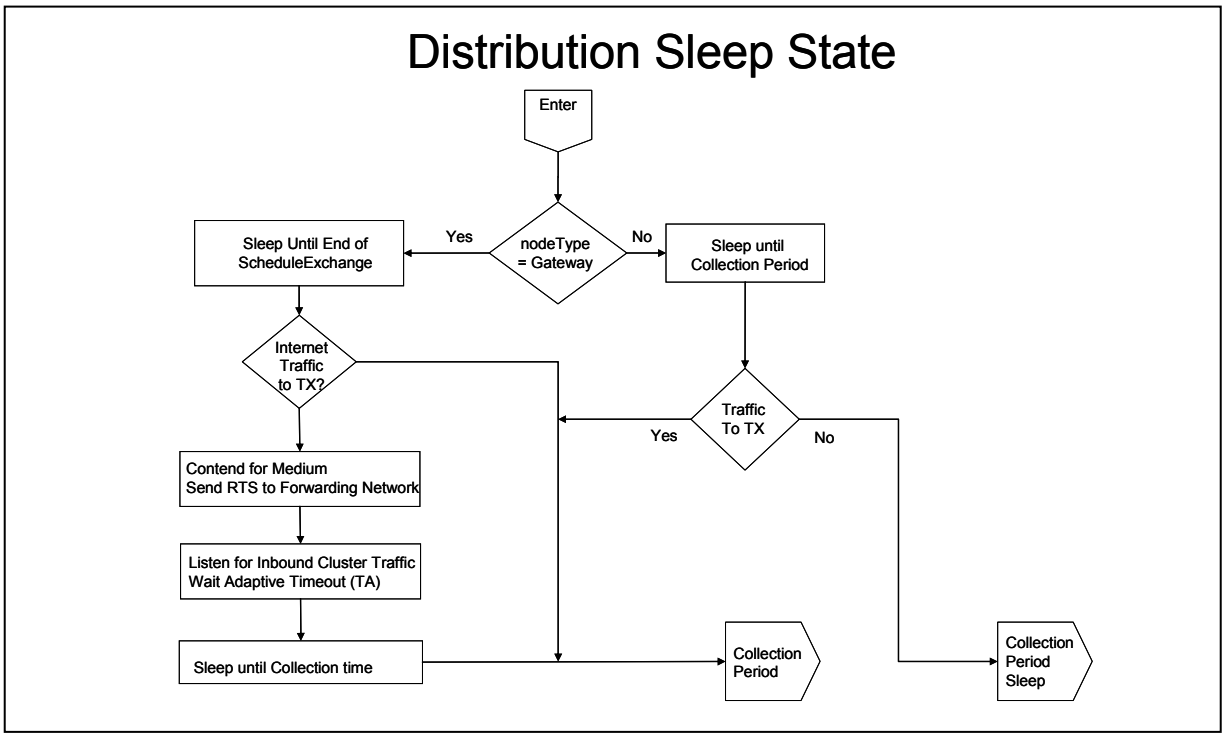

Figure A-4 GMAC Distribution Sleep State Flow Diagram

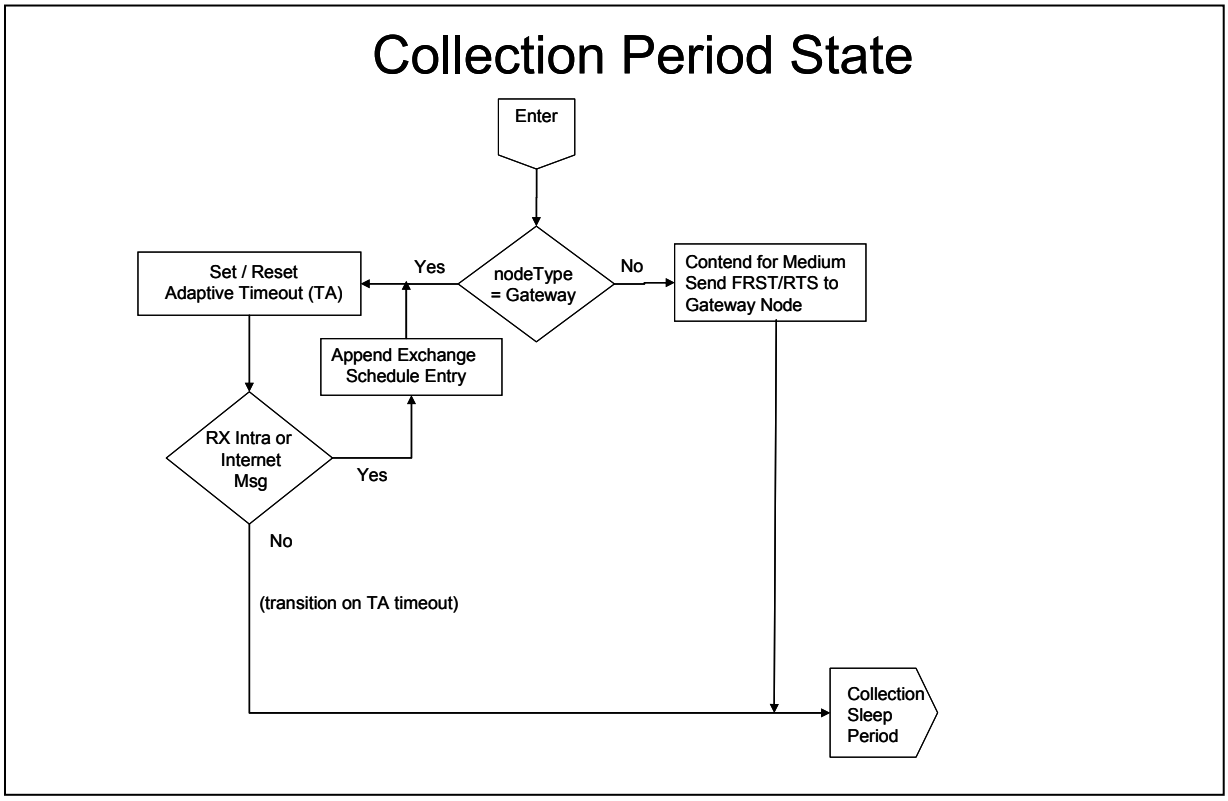

Figure A-5 GMAC Collection State Flow Diagram

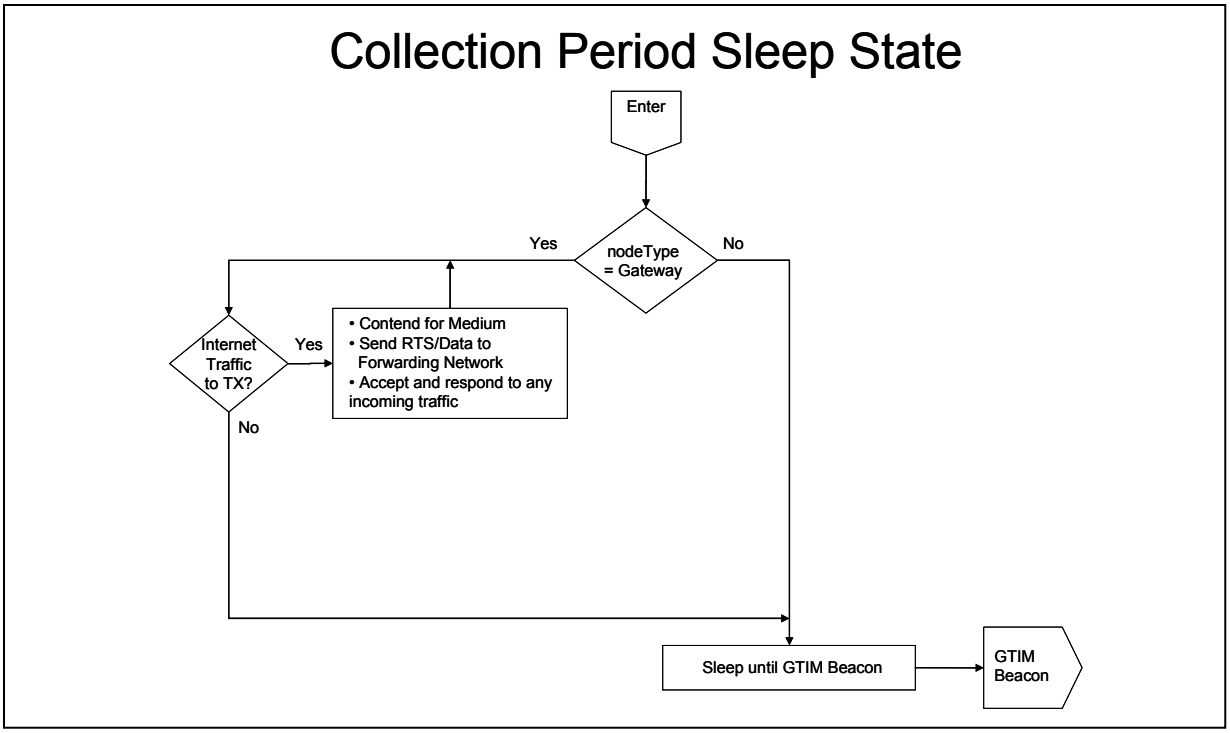

Figure A-6 GMAC Collection Sleep State Flow Diagram

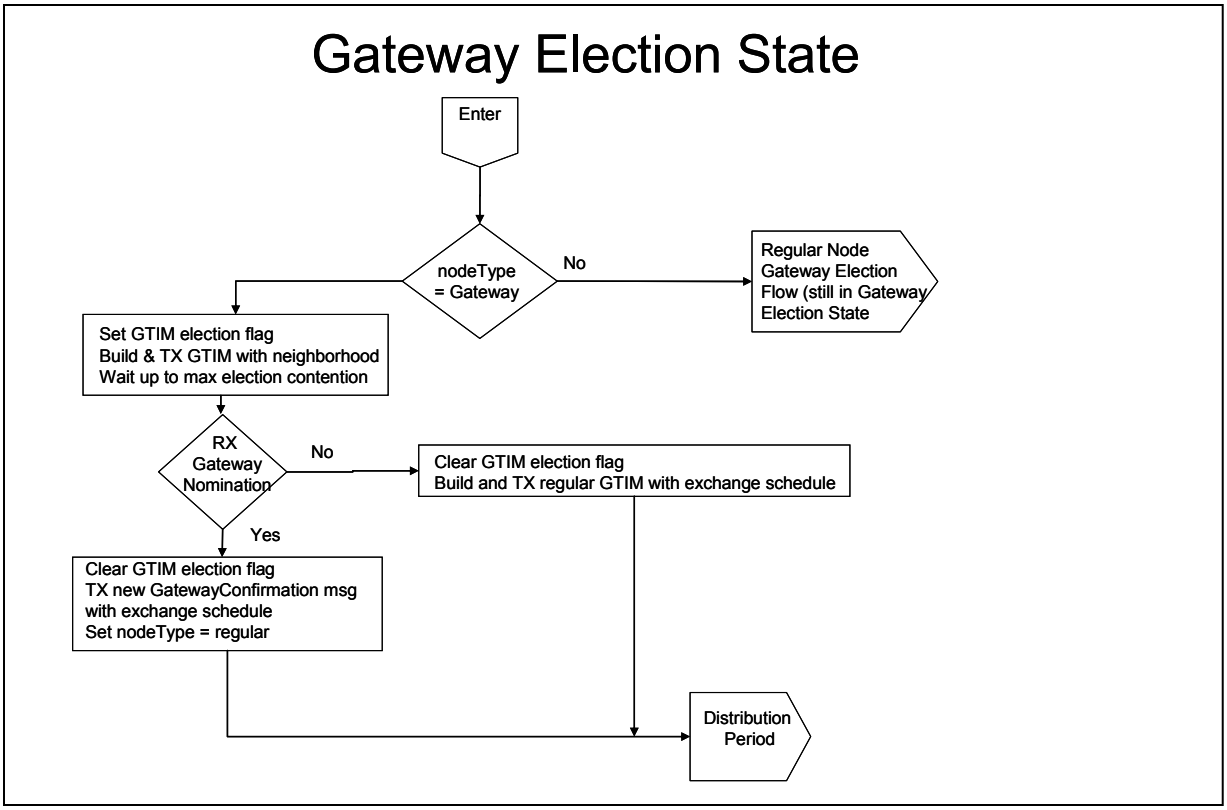

Figure A-7 GMAC Election State Flow Diagram

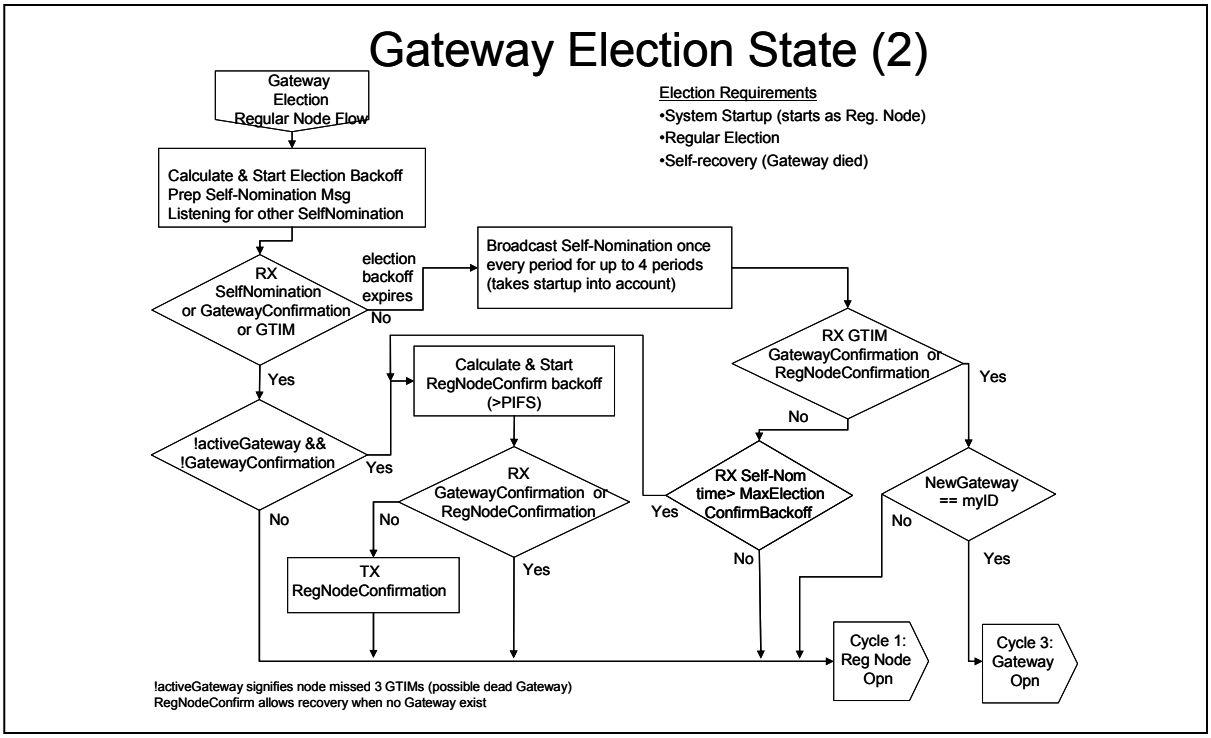

Figure A-8 GMAC Election State Flow Diagram (2)

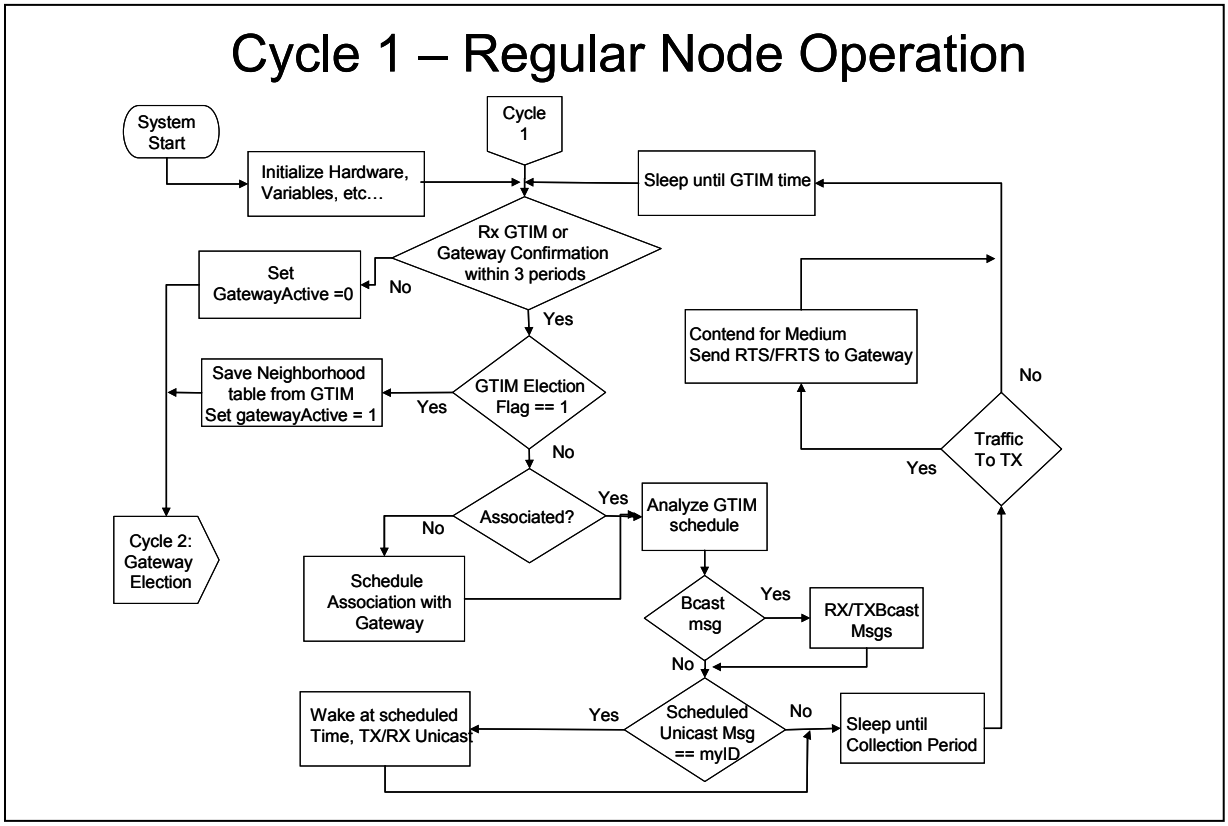

Figure A-9 GMAC Regular Node Flow Diagram

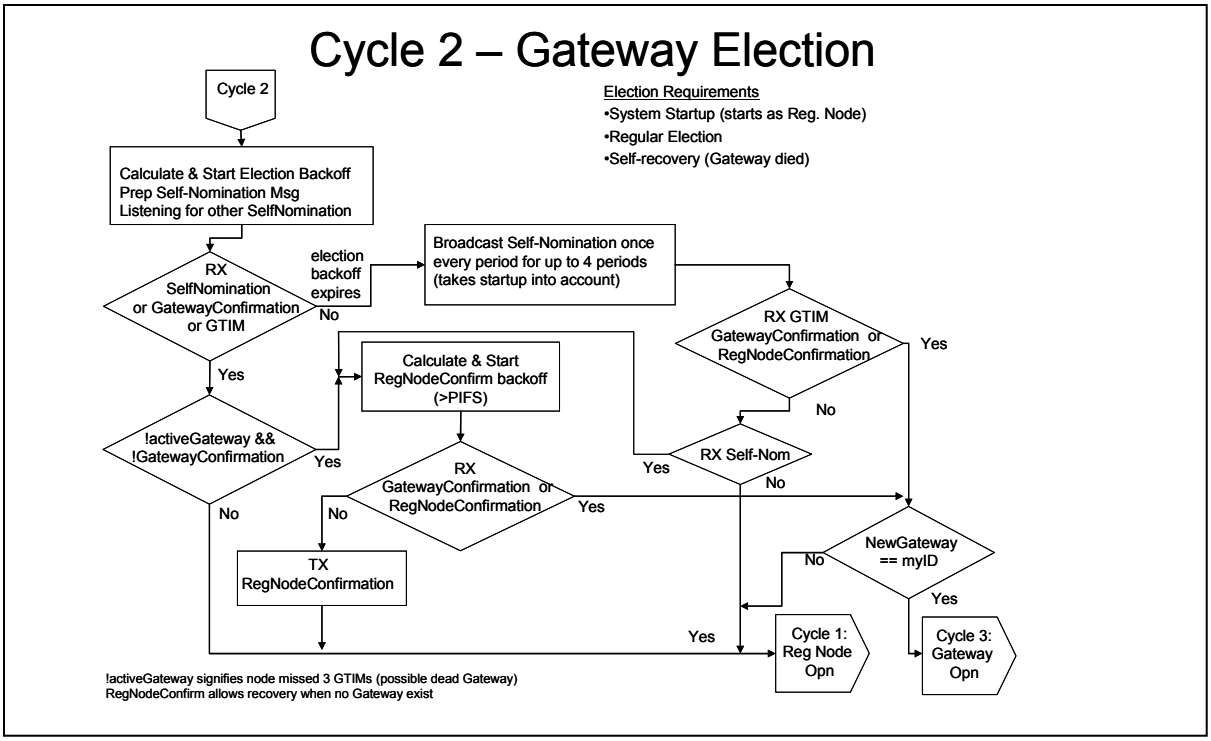

Figure A-10 GMAC Gateway Election Flow Diagram

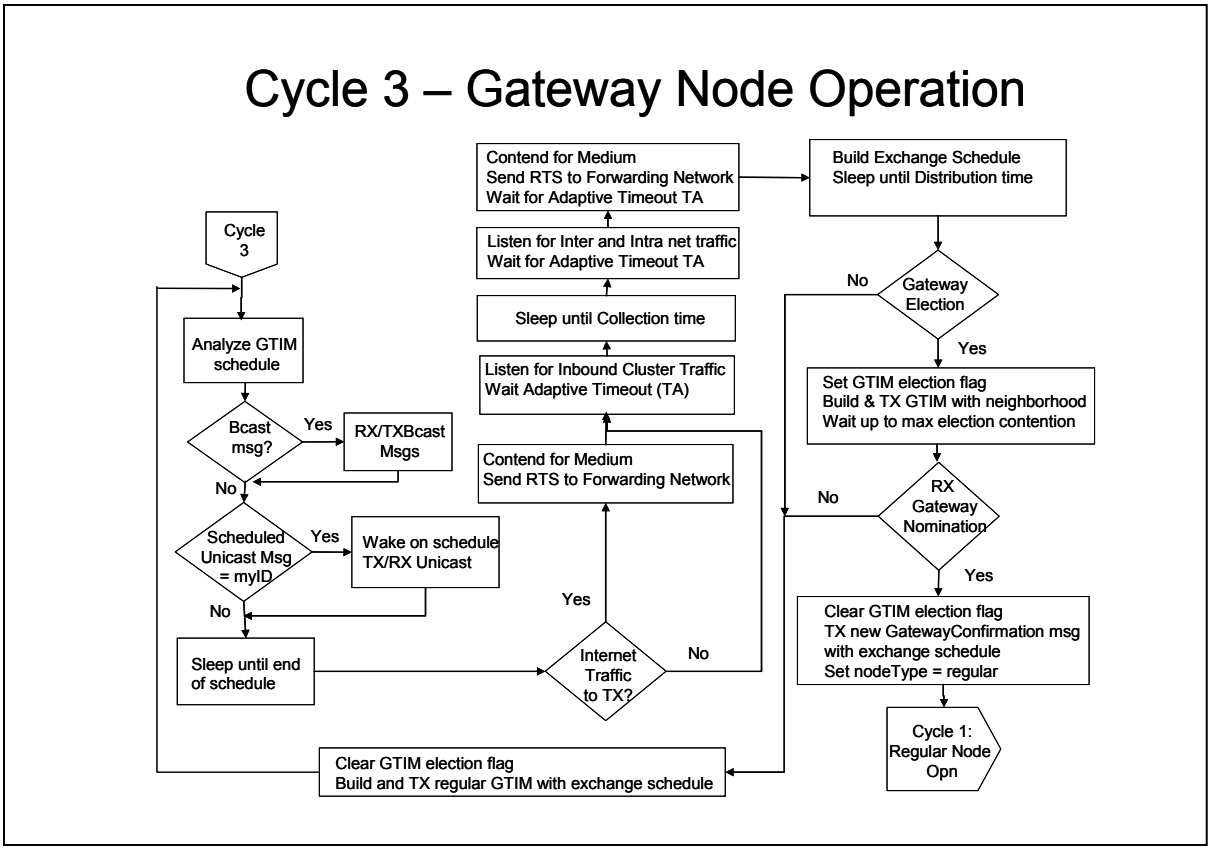

Figure A-11 GMAC Gateway Node Flow Diagram

Appendix B: IEEE 802.11 Power Save and GMAC Broadcast Message Comparisons

Table B-1 WSN Ad Hoc Environment Network Lifetime

	Broadcast Packet Rate					
Protocol	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	0 pkt/s
802.11 Ad Hoc PS	5.8 days	5.8 days	5.8 days	5.8 days	5.8 days	79 days
802.11 DCF PS	41 days	66 days	105 days	155 days	197 days	248 days
GMAC (Gateway Awake)	64 days	75 days	90 days	113 days	151 days	237 days

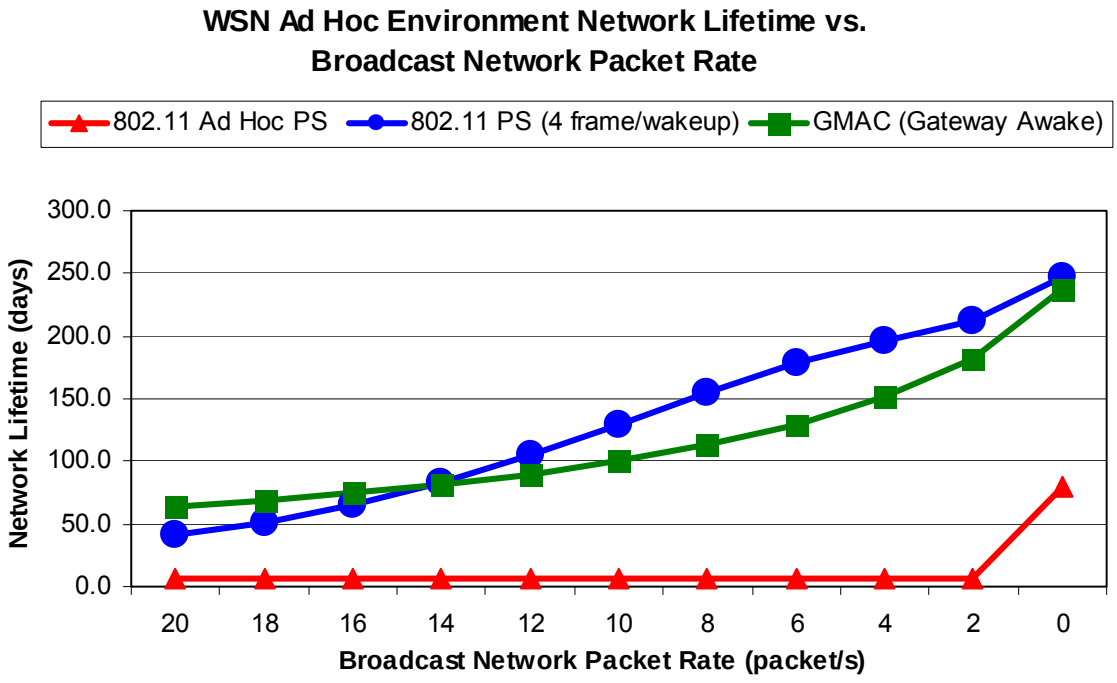

Figure B-1 802.11 and GMAC Network Lifetime in WSN Ad Hoc Environment

Table B-2 WLAN Infrastructure Environment Network Lifetime

	Broadcast Packet Rate					
Protocol	20 pkt/s	16 pkt/s	12 pkt/s	8 pkt/s	4 pkt/s	0 pkt/s
802.11 Ad Hoc PS	6.0 days	6.0 days	5.9 days	5.9 days	5.9 days	109 days
802.11 DCF PS	48 days	86 days	165 days	331 days	611 days	1694 days
GMAC (Gateway Awake)	81 days	101 days	131 days	186 days	317 days	1293 days

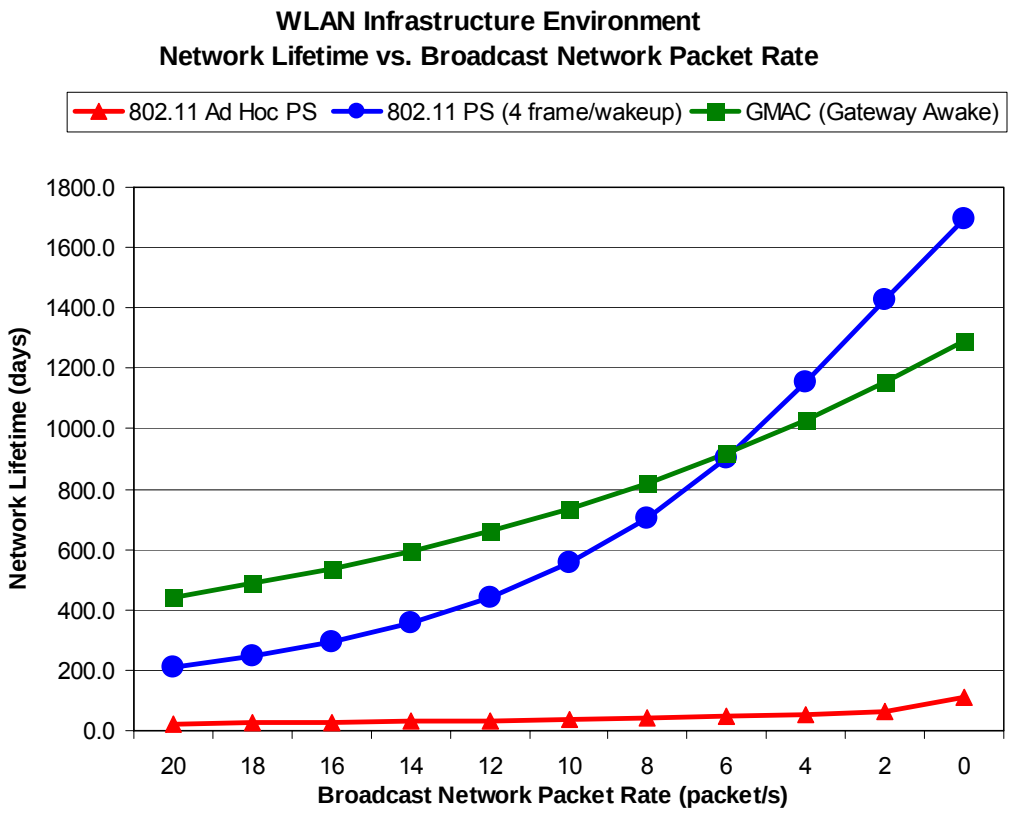

Figure B-2 802.11 and GMAC Network Lifetime in WLAN Infrastructure Environment

Appendix C: Instrumentation Circuit Calibration

Sensor Voltage Gain Calculations

Purpose

In order to perform more accurate and decisive testing simulations, an amplifier circuit was built to increase the voltages displayed on our oscilloscope. Using the magnified voltages allowed for ease in voltage change distinction. After results were obtained, however, output had to be converted back to scale. To do so, the amplifier's voltage and current gains had to be calculated.

Sink Resistor		R2 Resistor		Op Resistor	
Ideal	= 150 Ω	Ideal	= 1 Ω	Ideal	= 1 KΩ
Measure	= 148.426 KΩ	Measured	= 1.079 Ω	Measured	= 0.9890 KΩ

Ideal

Vi = 3V * R2 / (R2 + Sink Resistor)
= 3V * 1 Ω / (1 Ω + 150 Ω)
= 0.01987 V

Voltage Gain = 5 + 80KΩ / 1KΩ = 85

Current Gain = Voltage Gain / 1 Ω = 85

Calculated

Vi = 3V * R2 / (R2 + Sink Resistor)
= 3V * 1.079 Ω / (1.079 Ω + 148.426 Ω)
= 0.02165 V

Voltage Gain = 5 + 80KΩ / 0.9890KΩ = 85.89

Current Gain = Voltage Gain / 1.079 Ω = 79.60

Measured

Vi = 0.01938 V

Vout = 1.653 V

Voltage Gain = Vout / Vi = 85.294

Current Gain = Voltage Gain / 1.079 Ω = 79.049

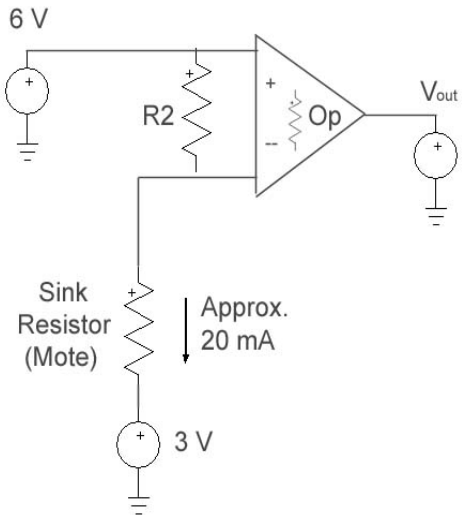

Figure C-1 Equivalent Amplification

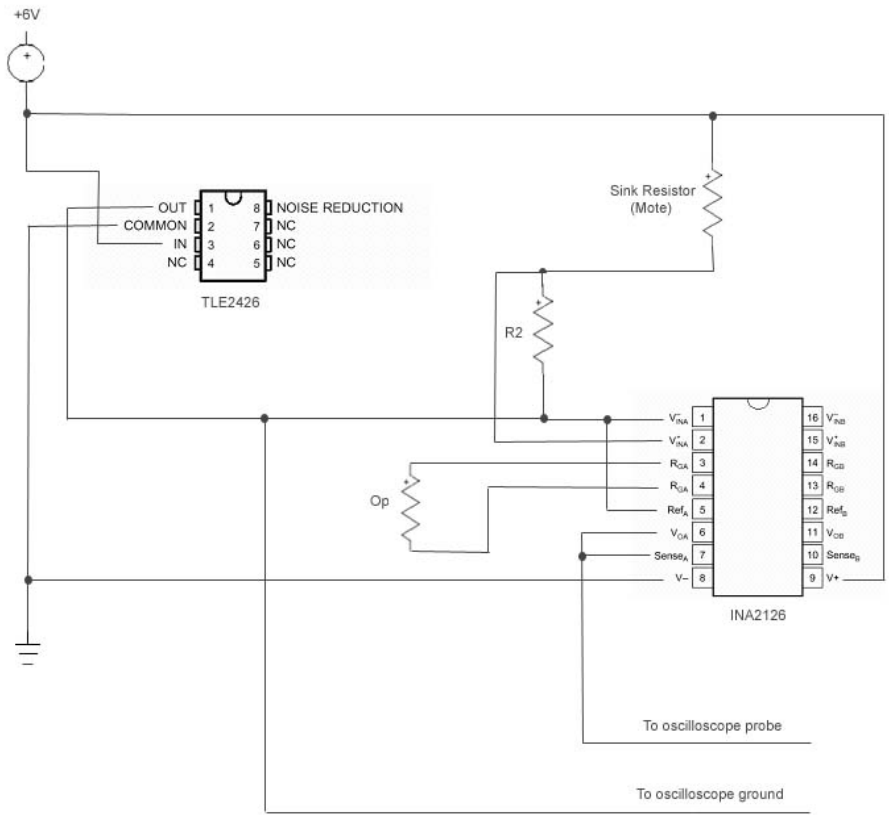

Figure C-2 Current Measuring Instrumentation Circuit

Appendix D: Iterative Schedule Parsing Scheme

```
Initial Loop
Set NumPackets
iteration = 0;

Every Loop
Sleep_duration = 0;  // reset sleep duration
for i = iteration to NumPackets
        {
        If src and dst !myAddress then
                {
                Sleep_duration =+ offset
                iteration++;
                }

        else if src == myAddress then
                {
                If sleep_duration != 0
                        {
                         Sleep (sleep_duration);
                        Set selfInterrupt for sleep_duration;
                        Break out of "for loop"
                        }
                Send_GTIM_Scheduled_Packet;
                Set selfInterrupt until next offset;
                iteration++;
                Break out of "for loop"
                }

        else dst == myAddress then
                {
                If sleep_duration != 0
                        {
                         Sleep (sleep_duration);
                        Set selfInterrupt for sleep_duration;
                        Break out of "for loop"
                        }
                // pause parsing to receive a packet
                Set selfInterrupt until next offset;
                iteration++;
                Break out of "for loop"
                }
        }
```

Appendix E: Analytical WSN MAC Protocol Models

Network Lifetime Calculations & Algorithms

Symbols:

Lf : Lifetime (sec) E' : energy / voltage (A*sec) C = Battery Capacity (A*sec)

sn: sensor node gw: gateway node

I_{Rx}: Receive Current (A) I_{Tx}: Transmit Current (A) I_{SLEEP} = LPM3 Sleep Current (A)

E'_{LPM3}: Ave energy/voltage for LPM3 Transition (A*s)

$E'_{Election}$: Ave election energy/voltage (A*s)
(reduces network lifetime by less than 0.004% for 6 hour election period)

T_{GTIM}: Gateway TIM time T_{Frame}: Protocol Frame Time (500ms)

T_{Sleep}: Sleep Time T_{LPM3}: LPM3 Transition Time

$T_{FD} = T_{FRAME * DUTY}$ T_{Data}: Data Packet = 8/BR * PKTS

T_{Cont}: Contention time (DIFS and Ave Contention Backoff = 560µs)

Tn: transmitting node Rn: Receiving node

oth: Other nodes PKTS: packets/frame

NetworkAve: Network Weighted Average

N: Number of Nodes in Network

PKTSUM: ∑(from n = 1 to n = PKTS-1) n = PKTS * (PKTS-1) / 2

T_{TA}: Adaptive Timeout = 1.5 * (DIFS + Ave Contention + RTS+ SIFS)

GMAC MODEL

GMAC Election Model

Gateway Node: start election period, collect self-nomination pkt (receive contention, receive SelfNom, receive a SIFS), and transmit a Confirmation pkt.

$E'(t)_{gw} = ((T_{CONT} + T_{SelfNom} + T_{SIFS}) * I_{RX}) + (T_{Confirm} * I_{TX})$

New Gateway Node: start election period, transmit self-nomination pkt (receive contention, transmit SelfNom) receive a SIFS, and receive a Confirmation Pkt.

$E'(t)_{new\_gw} = (T_{CONT} * I_{RX}) + (T_{SelfNom} * I_{TX}) + (T_{SIFS} + T_{Confirm}) * I_{RX}$

Other Nodes: start election period, receive a Self-Nomination pkt (receive contention, receive SelfNom), receive a SIFS, and receive a Confirmation Pkt.

$E'(t)_{oth} =((T_{CONT} + T_{SelfNom} + T_{SIFS} + T_{Confirm}) * I_{RX}) + (T_{Confirm} * I_{TX})$

Entire Network: one Gateway node, one new gateway node, and N – 2 other nodes.

$E'(t)_{NetworkAve} = [E'(t)_{gw} + E'(t)_{new\_gw} + (N-2) * E'(t)_{oth}] / N$

Per frame cost = $E'(t)_{NetworkAve}$ / (ave number of frames per election)

Average number of frames per election = (6 hours * 3600 sec/hr) / (0.500 sec/frame) = 43200 frames per election

GMAC No Traffic Model

Gateway Node: start collection period, wait for time out period, sleep, wake up, transmit a GTIM, wait for time out period, go back to sleep, and wakeup for next collection period.

$E'(t)_{gw} = (T_{GTIM} * I_{TX}) + (2* T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + 2E'_{LPM3} + E'_{Election}$
where $T_{Sleep} = T_{FRAME} - (T_{TA} + T_{GTIM} + T_{LPM3})$

Other Nodes: sleep, wake up SIFS early, receive GTIM, and return to sleep.

$E'(t)_{oth} = [(T_{SIFS} + T_{GTIM}) * I_{RX})] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3} + E'_{Election}$
where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + 2T_{LPM3})$

Entire Network: one Gateway node and N – 1 Other nodes.

$E'(t)_{NetworkAve} = [E'(t)_{gw} + (N-1) * E'(t)_{oth}] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

GMAC Unicast Traffic Model

Gateway Node: start collection period, collect #packets FRTSs (receive contention, receive the FRTS, receive a SIFS, transmit an ACK), wait for time out period, sleep, wake up, transmit a GTIM, go back to sleep, and wakeup for next collection period.

$E'(t)_{gw} = [(T_{CONT} + T_{FRTS} + T_{SIFS)} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + (2T_{TA} * I_{RX}) + (T_{GTIM} * I_{TX}) + (T_{Sleep} * I_{SLEEP}) + 2E'_{LPM3} + E'_{Election}$

Where $T_{Sleep} = T_{FRAME} - ([(T_{CONT} + T_{FRTS} + T_{SIFS} + T_{ACK}) *PKTS] + 2T_{TA} + T_{GTIM} + T_{LPM3})$

Transmitting Node: start collection period, transmit #packets FRTS (receive contention, transmit a FRTS, receive a SIFS, receive an ACK), receive contention for the summation of #packets-1 (receive contention, receive a DIFS, receive a FRTS, receive a SIFS, receive an ACK), sleep, wake up SIFS before GTIM, receive GTIM, send #packets to receiving nodes (receive SIFS, transmit data packet, receive SIFS for data packet, receives ACK for data packet), and return to sleep.

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{FRTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS + [(T_{CONT} * I_{RX}) + (T_{FRTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTSUM + (T_{SIFS} * I_{RX} + T_{GTIM} * I_{RX}) + [(T_{SIFS} + I_{RX}) + (T_{DATA} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS + (T_{ST} * I_{SLEEP}) + (PKTS * E'_{LPM3}) + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + [(T_{CONT} + T_{FRTS} + T_{SIFS} + T_{ACK}) *PKTS)] + [(T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS] + (T_{LPM3} * PKTS))$

Receiving Node: wake up SIFS early, receive GTIM, receive # packets from transmitting nodes (receive SIFS, receive data packet, receive SIFS for data packet, transmit ACK for data packet), and return to sleep.

$E'(t)_{Rn} = (T_{SIFS} * I_{RX} + T_{GTIM} * I_{RX}) + [(T_{SIFS} + I_{RX}) + (T_{DATA} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + (T_{Sleep} * I_{SLEEP}) + (PKTS * E'_{LPM3}) + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + [(T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK} + T_{LPM3}) * PKTS])$

Other Nodes: sleep, wake up SIFS early, receive GTIM, and return to sleep.

$E'(t)_{oth} = [(T_{SIFS} + T_{GTIM}) * I_{RX})] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3} + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + T_{LPM3})$

Entire Network: one Gateway node, one Transmitter, one Receiver, and N – 3 Other nodes.

$E'(t)_{NetworkAve} = [E'(t)_{gw} + E'(t)_{Tn} + E'(t)_{Rn} + (N-3) * E'(t)_{oth}] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

GMAC Regular Broadcast Traffic Model

Gateway Node: start collection period, collect #packets FRTSs (receive contention, receive the FRTS, receive a SIFS, transmit an ACK), wait for time out period, sleep, transmit a GTIM, receive SIFS, receive broadcast PKT, and return to sleep.

$E'(t)_{gw} = [(T_{CONT} + T_{FRTS} + T_{SIFS)} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + (2T_{TA} * I_{RX}) + (T_{GTIM} * I_{TX}) + [(2T_{SIFS} + T_{DATA}) * I_{RX} * PKTS] + (T_{Sleep} * I_{SLEEP}) + 2E'_{LPM3} + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - ([(T_{CONT} + T_{FRTS} + T_{SIFS} + T_{ACK}) *PKTS] + 2T_{TA} + T_{GTIM} + [(T_{DATA} + 2T_{SIFS}) *PKTS] + 2T_{LPM3})$

Transmitting Node: start collection period, transmit #packets FRTS (receive contention, transmit a FRTS, receive a SIFS, receive an ACK), receive message overhearing for the summation of #packets-1 (receive contention, receive a FRTS, receive a SIFS, receive an ACK), sleep, wake up SIFS early, receive GTIM, send #packets to all nodes (receive SIFS, transmit data packet), and return to sleep.

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{FRTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS + [(T_{CONT} * I_{RX}) + (T_{FRTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTSUM + (T_{SIFS} * I_{RX} + T_{GTIM} * I_{RX}) + [(T_{SIFS} + I_{RX}) + (T_{DATA} * I_{TX})] * PKTS + (T_{ST} * I_{SLEEP}) + (2 * E'_{LPM3}) + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + [(T_{CONT} + T_{FRTS} + T_{SIFS} + T_{ACK}) *(PKTS+ PKTSUM)] + [(T_{SIFS} + T_{DATA}) * PKTS] + 2T_{LPM3})$

Receiving Node (All others): wake up SIFS early, receive GTIM, receive #packets from transmitting nodes (receive SIFS, receive data packet), and return to sleep.

$E'(t)_{Rn} = (T_{SIFS} * I_{RX} + T_{GTIM} * I_{RX}) + [(T_{SIFS} + I_{RX}) + (T_{DATA} * I_{RX})] * PKTS + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3} + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + [(T_{SIFS} + T_{DATA}) * PKTS] + T_{LPM3})$

Entire Network: one Gateway node, one Transmitter, and N – 2 Receivers.

$E'(t)_{NetworkAve} = [E'(t)_{gw} + E'(t)_{Tn} + (N-2) * E'(t)_{Rn}] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

GMAC Rogue Denial of Sleep Broadcast Model

Gateway Node: start collection period, collect #packets FRTSs (receive contention, receive the FRTS, receive a SIFS, transmit an ACK), wait for time out period, sleep, transmit a GTIM, and return to sleep.

$E'(t)_{gw} = [(T_{CONT} + T_{FRTS} + T_{SIFS)} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + (2T_{TA} * I_{RX}) + (T_{GTIM} * I_{TX}) + (T_{Sleep} * I_{SLEEP}) + 2E'_{LPM3} + E'_{Election}$

where $T_{Sleep} = T_{FRAME} - ([(T_{CONT} + T_{FRTS} + T_{SIFS} + T_{ACK}) *PKTS] + 2T_{TA} + T_{GTIM} + 2T_{LPM3})$

Other Nodes: sleep, wake up SIFS early, receive GTIM, and return to sleep.

$E'(t)_{oth} = [(T_{SIFS} + T_{GTIM}) * I_{RX})] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3} + E'_{Election}$
where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + T_{LPM3})$

Entire Network: one Gateway node and N-1 Other nodes

$E'(t)_{NetworkAve} = [E'(t)_{gw} + (N-1) * E'(t)_{oth}]/ N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

SMAC Model

SMAC No Traffic Model

$T_{Duty:}$ Active period of frame cycle (10% of T_{FRAME})

All Nodes: receive for FRAME*DUTY and return to sleep.

$E'(t)_{NetworkAve} = (T_{Duty} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{Duty} + T_{LPM3})$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

SMAC Unicast Traffic Model

Transmitting Node: transmit #packets (receive contention, transmit a RTS, receive a SIFS, receive an CTS, receive a SIFS, transmit data packet, receive a SIFS, receive an ACK), remain in active mode until (FRAME*DUTY) is complete, and return to sleep.

$T_{TnTransmissionSequence} = (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS$

$= (T_{CONT} + T_{RTS} + T_{3SIFS} + T_{CTS} + T_{DATA} + T_{ACK}) * PKTS$

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{CTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{DATA} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS + [(T_{Duty} - T_{TnTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{Duty} + T_{LPM3})$

Receiving Node: receive #packets (receive contention, receive a DIFS, receive a RTS, receive a SIFS, transmit a CTS, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), remain in active mode until (FRAME*DUTY) is complete, and return to sleep.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS$

$= (T_{CONT} + T_{RTS} + T_{3SIFS} + T_{CTS} + T_{DATA} + T_{ACK}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{CTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{DATA} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + [(T_{Duty} - T_{RnTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{Duty} + T_{LPM3})$

Other Nodes: overhear initial reservation for #packets for NAV sleep (receive contention, receive a RTS, go to sleep throughout NAV duration), remain in active mode until (FRAME*DUTY) is complete, and return to sleep.

$T_{OthTransmissionSequence} = (T_{CONT} + T_{RTS}) * PKTS$

$E'(t)_{Oth} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{RX})] * PKTS + [(T_{Duty} - T_{OthTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = (T_{FRAME} – T_{Duty}) + (T_{TnTransmissionSequence} – T_{OthTransmissionSequence} - T_{LPM3}) * PKTS$

note: $(T_{TnTransmissionSequence} – T_{OthTransmissionSequence} - T_{LPM3})$ must be greater than 0 to include as T_{Sleep}

Entire Network: one Transmitter, one Receiver, and N – 2 Other nodes

$E'(t)_{NetworkAve} = [\ E'(t)_{Tn} + E'(t)_{Rn} + (N-2) * E'(t)_{oth}\ ]/ N$

$Lf(t)_{ave} = (Capacity\ * T_{Frame}) / E'(t)_{NetworkAve}$

SMAC Regular Broadcast Traffic Model

Transmitting Node: Transmit #packets (receive contention, transmit data packet), remain in active mode until (FRAME*DUTY) is complete, and return to sleep.

$T_{TnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{TX})\ ] * PKTS + [(T_{Duty} - T_{TnTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$
where $T_{Sleep} = T_{FRAME} - (T_{Duty} + T_{LPM3})$

Receiving Node: Receive #packets (receive contention, receive a DIFS, receive a data packet), remain in active mode until (FRAME*DUTY) is complete, and return to sleep.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{RX})\ ] * PKTS + [(T_{Duty} - T_{TnTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$
where $T_{Sleep} = T_{FRAME} - (T_{Duty} + T_{LPM3})$

Entire Network: one Transmitter and N – 1 Receiver nodes.

$E'(t)_{NetworkAve} = [\ E'(t)_{Tn} + (N-1) * E'(t)_{Rn}\ ]/ N$

$Lf(t)_{ave} = (Capacity\ * T_{Frame}) / E'(t)_{NetworkAve}$

SMAC Rogue Denial of Sleep Broadcast Model

Receiving Node: Receive #packets (receive contention, receive a DIFS, receive a data packet), remain in active mode until (FRAME*DUTY) is complete, and return to sleep.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{RX})\ ] * PKTS + [(T_{Duty} - T_{TnTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$
where $T_{Sleep} = T_{FRAME} - (T_{Duty} + T_{LPM3})$

Entire Network: N Receiver nodes.

$E'(t)_{NetworkAve} = E'(t)_{Rn}$

$Lf(t)_{ave} = (Capacity\ * T_{Frame}) / E'(t)_{NetworkAve}$

TMAC Model

TMAC + No Traffic

Nodes: Receive mode for a timeout period and return to sleep.

$E'(t)_{NetworkAve} = (T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{TA} + T_{LPM3})$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

TMAC + Unicast Traffic Model

Transmitting Node: Transmit #packets (receive contention, transmit a RTS, receive a SIFS, receive an CTS, receive a SIFS, transmit Data Packet, receive a SIFS, receive an ACK), remain in active mode until a timeout period has elapsed, and return to sleep.

$T_{TnTransmissionSequence} = (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS$
$= (T_{CONT} + T_{RTS} + T_{3SIFS} + T_{CTS} + T_{DATA} + T_{ACK}) * PKTS$

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{CTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{DATA} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS + (T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{TnTransmissionSequence} + T_{TA} + T_{LPM3})$

Receiving Node: receive #packets (receive contention, receive a DIFS, receive a RTS, receive a SIFS, transmit a CTS, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), remain in active mode until a timeout period has elapsed, and return to sleep.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS$
$= (T_{CONT} + T_{RTS} + T_{3SIFS} + T_{CTS} + T_{DATA} + T_{ACK}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{CTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{DATA} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + (T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{TnTransmissionSequence} + T_{TA} + T_{LPM3})$

Other Nodes: overhear initial reservation for #packets for NAV sleep (receive contention, receive a RTS, go to sleep throughout NAV duration), remain in active mode until TA timeout period is complete, and return to sleep.

$T_{OthTransmissionSequence} = (T_{CONT} + T_{RTS}) * PKTS$

$E'(t)_{Oth} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{RX})] * PKTS + [(T_{Duty} - T_{OthTransmissionSequence}) * I_{RX}] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} – T_{TA} - T_{OthTransmissionSequence}$

provided that $(T_{TnTransmissionSequence} – T_{OthTransmissionSequence} - T_{LPM3})$ is greater than 0

Otherwise $T_{Sleep} = T_{FRAME} – (T_{TA} + T_{TnTransmissionSequence})$

Entire Network: one Transmitter, one Receiver, and N – 2 Other nodes.

$E'(t)_{NetworkAve} = [E'(t)_{Tn} + E'(t)_{Rn} + (N-2) * E'(t)_{oth}]/ N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

<u>TMAC + Regular Broadcast Traffic Model</u>

Transmitting Node: transmit #packets (receive contention, receive a DIFS, transmit data packet), remain in active mode until a timeout period has elapsed, and return to sleep.

$T_{TnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{TX}) * PKTS + (T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$
where $T_{Sleep} = T_{FRAME} - (T_{TnTransmissionSequence} + T_{TA} + T_{LPM3})$

Receiving Node: Receive #packets (receive contention, receive a data packet), remain in active mode until a timeout period has elapsed, and return to sleep.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{RX})]* PKTS + (T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - (T_{TnTransmissionSequence} + T_{TA} + T_{LPM3})$

Entire Network: one Transmitter and N-1 Receiver nodes.

$E'(t)_{NetworkAve} = [E'(t)_{Tn} + (N-1) * E'(t)_{Rn}]/ N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

<u>TMAC + Rogue Denial of Sleep Broadcast Model</u>

Receiving Node: Receive #packets (receive contention, receive a Data Packet), remain in active mode until a timeout period has elapsed, and return to sleep.

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{RX})]* PKTS + (T_{TA} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$

where $T_{Sleep} = T_{FRAME} - ((T_{CONT} + T_{DATA}) * PKTS + T_{TA} + T_{LPM3})$

Entire Network: N Receiver nodes.

$E'(t)_{NetworkAve} = E'(t)_{Rn}$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Model (Active Mode)

IEEE 802.11 No Traffic Model

All Nodes: duty cycle 100%

$E'(t)_{NetworkAve} = T_{FRAME} * I_{RX}$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Unicast Traffic Model

Transmitting Node: transmit #packets (receive contention, transmit a RTS, receive a SIFS, receive a CTS, receive a SIFS, transmit data packet, receive a SIFS, receive an ACK), and remain in active mode.

$T_{TnTransmissionSequence} = (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS$

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{CTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{DATA} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS + [(T_{FRAME} - T_{TnTransmissionSequence}) * I_{RX}]$

Receiving Node: receive #packets (receive contention, receive a RTS, receive a SIFS, transmit a CTS, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), and remain in active mode.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{CTS} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{DATA} * I_{RX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + [(T_{FRAME} - T_{RnTransmissionSequence}) * I_{RX}]$

Entire Network: One Transmitter, (N – 1) Receivers

$E'(t)_{NetworkAve} = [E'(t)_{Tn} + (N-1) * E'(t)_{Rn}]/ N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Regular Broadcast Traffic Model

Transmitting Node: transmit #packets (receive contention, transmit a data packet), and remain in active mode.

$T_{TnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Tn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{TX})] * PKTS + [(T_{FRAME} - T_{TnTransmissionSequence}) * I_{RX}]$

Receiving Node: receive #packets (receive contention, receive a DIFS, and receive a data Packet) and remain in active mode.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{RX})] * PKTS + [(T_{FRAME} - T_{RnTransmissionSequence}) * I_{RX}]$

Entire Network: one Transmitter and (N – 1) Receiver nodes.

$E'(t)_{NetworkAve} = [E'(t)_{Tn} + (N-1) * E'(t)_{Rn}]/ N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Rogue Denial of Sleep Broadcast Model

All Nodes: receive #packets (receive contention, receive a DIFS, and receive a data packet) and remain in active mode.

$T_{RnTransmissionSequence} = (T_{CONT} + T_{DATA}) * PKTS$

$E'(t)_{Rn} = [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{RX})] * PKTS + [(T_{FRAME} - T_{RnTransmissionSequence}) * I_{RX}]$

Entire Network: N Receiver nodes

$E'(t)_{NetworkAve} = E'(t)_{Rn}$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Infrastructure Power Save Mode

Assume that the power save BTIM cycle is 4 cycles.

IEEE 802.11 Infrastructure PS Mode No Traffic Model

Access Point: transmit BSS beacon with TIM and monitor network.

$E'(t)_{ap} = (T_{BTIM} * I_{TX}) + (T_{FRAME} - T_{BTIM}) * I_{RX}$

Other Nodes: sleep, wake up SIFS early on every fourth beacon period, receive BTIM, and return to sleep.

$E'(t)_{oth} = [(T_{SIFS} + T_{BTIM})/4 * I_{RX})] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}/4$
where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{BTIM} + T_{LPM3})/4$

Entire Network: one AP node and N – 1 other nodes.

$E'(t)_{NetworkAve} = [E'(t)_{ap} + (N-1) * E'(t)_{oth}] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Infrastructure PS Mode Unicast Traffic Model

AP Node: start period, transmit BSS beacon with TIM, send PS poll packets #packets (receive contention, receive the PS Poll, receive a SIFS, transmit a data packet, receive a SIFS, receive an ACK), collect #packets (receive contention, receive the RTS, receive a SIFS, transmit a CTS, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), and monitor the network.

$E'(t)_{ap} = (T_{BTIM} * I_{TX}) + [((T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS}) * I_{RX}) + (T_{DATA} * I_{TX}) + ((T_{SIFS} + T_{ACK}) * I_{RX})] * PKTS + [((T_{CONT} + T_{RTS} + T_{SIFS}) * I_{RX}) + (T_{CTS} * I_{TX}) + ((T_{SIFS} + T_{DATA} + T_{SIFS}) * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS + (Frame - T_{BTIM} - T_{TnTransmissionSequence}) * I_{RX}$

where $T_{TnTransmissionSequence} = [(T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS] + [(T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS]$

Transmitting Node: wake up SIFS before BTIM, receive BTIM, overhear other PS-Poll transmissions for the #packets (receive contention, receive the PS Poll, receive a SIFS, receive a data packet, receive a SIFS, receive an ACK), send #packets to AP (receive contention, transmit a RTS, receive a SIFS, receive a CTS, transmit data packet, receive a SIFS, receive an ACK), overhear other transmissions for the summation of #packets-1 (receive contention, receive a RTS, receive a SIFS, receive a CTS, receive data packet, receive a SIFS, receive an ACK), and return to back to sleep.

$E'(t)_{Tn} = (T_{SIFS} + T_{BTIM}) * I_{RX} + [(T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * I_{RX}] * PKTS$
$+ [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{TX}) + (T_{SIFS} + T_{CTS} + T_{SIFS}) * I_{RX} + (T_{DATA} * I_{TX}) + ((T_{SIFS} + T_{ACK}) * I_{RX}] * PKTS$
$+ [(T_{CONT}\ T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * I_{RX}] * PKTSUM$
$+ E'_{LPM3} + (Frame - T_{BTIM} - T_{LPM3} - T_{TnTransmissionSequence}) * I_{RX}$

where $T_{TnTransmissionSequence} = [(T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS]$
$+ [(T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * (PKTS + PKTSUM)]$

Receiving Node: wake up SIFS before BTIM, receive BTIM, overhear other PS-Poll transmissions for the summation of #packets-1 (receive contention, receive the PS Poll, receive a SIFS, receive a data packet, receive a SIFS, receive an ACK), send #packets PS-Poll to AP (receive contention, transmit the PS Poll, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), and return to back to sleep.

$E'(t)_{Tn} = (T_{SIFS} + T_{BTIM}) * I_{RX} + [(T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * I_{RX}] * PKTSUM$
$+ [(T_{CONT} * I_{RX}) + (T_{PS\text{-}Poll} * I_{TX}) + (T_{SIFS} + T_{DATA} + T_{SIFS}) * I_{RX} + (T_{ACK} * I_{TX})] * PKTS$
$+ E'_{LPM3} + (Frame - T_{BTIM} - T_{LPM3} - T_{TnTransmissionSequence}) * I_{RX}$

where $T_{TnTransmissionSequence} = (T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * (PKTS+ PKTSUM)$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + [(T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK} + T_{LPM3}) * PKTS])$

Other Nodes: sleep, wake up SIFS early on every fourth beacon period, receive BTIM, and return to sleep.

$E'(t)_{oth} = [(T_{SIFS} + T_{BTIM})/4 * I_{RX})] + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}/4$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{BTIM} + T_{LPM3})/4$

Entire Network: One AP node, one Transmitter, one Receiver, N – 3 Other nodes

$E'(t)_{NetworkAve} = [E'(t)_{gw} + E'(t)_{Tn} + E'(t)_{Rn} + (N-3) * E'(t)_{oth}] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

<u>IEEE 802.11 Infrastructure PS Mode Regular Broadcast Traffic Model</u>

AP Node: start period, transmit BSS beacon with TIM, send DTIM broadcast messages (receive a SIFS, transmit a data packet), collect #packets (receive contention, receive the RTS, receive a SIFS, transmit a CTS, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), and monitor the network.

$E'(t)_{ap} = (T_{BTIM} * I_{TX}) + [(T_{SIFS} * I_{RX}) + (T_{DATA} * I_{TX})] * PKTS$
$+ [((T_{CONT} + T_{RTS} + T_{SIFS}) * I_{RX}) + (T_{CTS} * I_{TX}) + ((T_{SIFS} + T_{DATA} + T_{SIFS}) * I_{RX}) + (T_{ACK} * I_{TX})] * PKTS$
$+ (Frame - T_{BTIM} - T_{TnTransmissionSequence}) * I_{RX}$

where $T_{TnTransmissionSequence} = [(T_{SIFS} + T_{DATA}) * PKTS] + [(T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS]$

Transmitting Node: wake up SIFS before BTIM, receive BTIM, receive #packets DTIM broadcast data (receive SIFS, receive data packet), send #packets to AP (receive contention, transmit a RTS, receive a SIFS, receive a CTS, transmit data packet, receive a SIFS, receive an ACK), overhear other transmissions for the summation of #packets-1 (receive contention, receive a RTS, receive a SIFS, receive a CTS, receive data packet, receive a SIFS, receive an ACK), and return to back to sleep.

$E'(t)_{Tn} = (T_{SIFS} + T_{BTIM}) * I_{RX} + [(T_{CONT} + T_{SIFS} + T_{DATA}) * I_{RX}] * PKTS$
$+ [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{TX}) + (T_{SIFS} + T_{CTS} + T_{SIFS}) * I_{RX} + (T_{DATA} * I_{TX}) + ((T_{SIFS} + T_{ACK}) * I_{RX}] * PKTS$
$+ [(T_{CONT}\ T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * I_{RX}] * PKTSUM$
$+ E'_{LPM3} + (Frame - T_{LPM3} - T_{TnTransmissionSequence}) * I_{RX}$

where $T_{TnTransmissionSequence} = [(T_{CONT} + T_{SIFS} + T_{DATA}) * PKTS]$
$+ [(T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * (PKTS+ PKTSUM)]$

Receiving Node: wake up SIFS before BTIM, receive BTIM, overhear other PS-Poll transmissions for the summation of #packets-1 (receive contention, receive the PS Poll, receive a SIFS, receive a data packet, receive a SIFS, receive an ACK), send #packets PS-Poll to AP (receive contention, transmit the PS Poll, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), and return to back to sleep.

$E'(t)_{Tn} = (T_{SIFS} + T_{BTIM}) * I_{RX} + [(T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * I_{RX}] * PKTSUM$
$+ [(T_{CONT} * I_{RX}) + (T_{PS\text{-}Poll} * I_{TX}) + (T_{SIFS} + T_{DATA} + T_{SIFS}) * I_{RX} + (T_{ACK} * I_{TX})] * PKTS$
$+ E'_{LPM3} + (Frame - T_{LPM3} - T_{TnTransmissionSequence}) * I_{RX}$

where $T_{TnTransmissionSequence} = (T_{CONT} + T_{PS\text{-}Poll} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * (PKTS+ PKTSUM)$

where $T_{Sleep} = T_{FRAME} - (T_{SIFS} + T_{GTIM} + [(T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK} + T_{LPM3}) * PKTS])$

Other Nodes: sleep, wake up SIFS early on every fourth beacon period, receive BTIM, receive #packets DTIM broadcast data (receive SIFS, receive data packet), and return to sleep.

$E'(t)_{oth} = [(T_{SIFS} + T_{BTIM})/4 * I_{RX})] + [(T_{CONT} + T_{SIFS} + T_{DATA}) * I_{RX}] * PKTS + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}/4$

where $T_{Sleep} = T_{FRAME} - [(T_{CONT} + T_{SIFS} + T_{DATA})] * PKTS - (T_{SIFS} + T_{BTIM} + T_{LPM3})/4$

Entire Network: One AP node, #PKTS Transmitters, (N – #PKTS – 1) Other nodes

$E'(t)_{NetworkAve} = [E'(t)_{ap} + (PKTS) * E'(t)_{Tn} + (N\text{-}PKTS\text{-}1) * E'(t)_{oth}] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Ad Hoc Power Save Mode

Assume that the power save BTIM cycle is 4 cycles.

IEEE 802.11 Ad Hoc PS Mode No Traffic Model

TBTT "Access Point": wake up, receive contention backoff, transmit IBSS beacon, receive during ATIM window, and monitor network.

$E'(t)_{tbttAP} = (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{TX}) + (T_{ATIM\_WINDOW} * I_{RX}) + (T_{FRAME} - T_{CONT} - T_{IBTIM} - T_{ATIM\_WINDOW}) * I_{RX}$

Other Nodes: wake up, receive contention backoff, receive IBSS beacon, receive during ATIM window, and then return to sleep.

$E'(t)_{oth} = (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{TX}) + (T_{ATIM\_WINDOW} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$
where $T_{Sleep} = T_{FRAME} - T_{CONT} - T_{IBTIM} - T_{ATIM\_WINDOW} - T_{LPM3}$

Entire Network: one TBTT "AP" node and N – 1 other nodes

$E'(t)_{NetworkAve} = [\, E'(t)_{tbttAP} + (N-1) * E'(t)_{oth} \,] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

IEEE 802.11 Ad Hoc PS Mode Unicast Traffic Model

TBTT "Access Point": wake up, receive contention backoff, transmit IBSS beacon, receive during ATIM window, and monitor network.

$E'(t)_{tbttAP} = (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{TX}) + (T_{ATIM\_WINDOW} * I_{RX}) + (T_{FRAME} - T_{CONT} - T_{IBTIM} - T_{ATIM\_WINDOW}) * I_{RX}$

Transmitting Node: wake up, receive contention backoff, receive IBSS beacon, transmit #packets ATIM requests during ATIM window (receive contention, transmit ATIM, receive SIFS, receive ACK), receive the remainder of ATIM window, send #packets (receive contention, transmit a RTS, receive a SIFS, receive a CTS, receive a SIFS, transmit data packet, receive a SIFS, receive an ACK), overhear other transmissions for the summation of #packets-1 (receive contention, receive a RTS, receive a SIFS, receive a CTS, receive data packet, receive a SIFS, receive an ACK), and return to back to sleep.

$$
\begin{aligned}
E'(t)_{Tn} = {} & (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{RX}) \\
& + [(T_{CONT} * I_{RX}) + (T_{ATIM} * I_{TX}) + (T_{SIFS} + T_{ACK}) * I_{RX}] * PKTS \\
& + [T_{ATIM\_WINDOW} - (T_{CONT} - T_{ATIM} - T_{SIFS} - T_{ACK}) * PKTS] * I_{RX} \\
& + [(T_{CONT} * I_{RX}) + (T_{RTS} * I_{TX}) + (T_{SIFS} + T_{CTS} + T_{SIFS}) * I_{RX} + (T_{DATA} * I_{TX}) + (T_{SIFS} * I_{RX}) + (T_{ACK} * I_{RX})] * PKTS \\
& + [(T_{CONT}\ T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * I_{RX}] * PKTSUM \\
& + [(T_{FRAME} - T_{LPM3} - T_{TnTransmissionSequence}) * I_{Sleep}] + E'_{LPM3}
\end{aligned}
$$

$$
\begin{aligned}
T_{TnTransmissionSequence} = {} & T_{CONT} + T_{IBTIM} + T_{ATIM\_WINDOW} \\
& + (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * (PKTS + PKTSUM)
\end{aligned}
$$

Receiving Node: wake up, receive contention backoff, receive IBSS beacon, receive #packets ATIM requests during ATIM window (receive contention, receive ATIM packet, receive SIFS, transmit ACK), receive the remainder of ATIM window, receive #packets (receive contention, receive a RTS, receive a SIFS, transmit a CTS, receive a SIFS, receive a data packet, receive a SIFS, transmit an ACK), and return to back to sleep.

$$
\begin{aligned}
E'(t)_{Rn} = {} & (T_{CONT} + T_{IBTIM}) * I_{RX} \\
& + [(T_{CONT} + T_{ATIM} + T_{SIFS}) * I_{RX} + (T_{ACK} * I_{TX})] * PKTS \\
& + [T_{ATIM\_WINDOW} - (T_{CONT} - T_{ATIM} - T_{SIFS} - T_{ACK}) * PKTS] * I_{RX} \\
& + [(T_{CONT} + T_{RTS} + T_{SIFS}) * I_{RX} + (T_{CTS} * I_{TX}) + (T_{SIFS} + T_{DATA} + T_{SIFS}) * I_{RX} + (T_{ACK} * I_{TX})] * PKTS \\
& + [(T_{FRAME} - T_{LPM3} - T_{RnTransmissionSequence}) * I_{Sleep}] + E'_{LPM3}
\end{aligned}
$$

$$
\begin{aligned}
T_{RnTransmissionSequence} = {} & T_{CONT} + T_{IBTIM} + T_{ATIM\_WINDOW} \\
& + (T_{CONT} + T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}) * PKTS
\end{aligned}
$$

Other Nodes: wake up, receive contention backoff, receive IBSS beacon, receive during ATIM window, and then return to sleep.

$$E'(t)_{oth} = (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{TX}) + (T_{ATIM\_WINDOW} * I_{RX}) + (T_{Sleep} * I_{SLEEP}) + E'_{LPM3}$$

where $T_{Sleep} = T_{FRAME} - T_{CONT} - T_{IBTIM} - T_{ATIM\_WINDOW} - T_{LPM3}$

Entire Network: One AP node, #PKTS Transmitters, #PKTS Receivers, N – (2 * #PKTS) – 1 Other nodes

$$E'(t)_{NetworkAve} = [\, E'(t)_{tbttAP} + (\#PKTS * E'(t)_{Tn}) + (\#PKTS * E'(t)_{Rn}) + (N - \#PKTS - \#PKTS - 1) * E'(t)_{oth} \,] / N$$

$$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$$

IEEE 802.11 Ad Hoc PS Mode Regular Broadcast Traffic Model

TBTT "Access Point": wake up, receive contention backoff, transmit IBSS beacon, receive during ATIM window, and monitor network (receive all broadcast messages).

$$E'(t)_{tbttAP} = (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{TX}) + (T_{ATIM\_WINDOW} * I_{RX}) + (T_{FRAME} - T_{CONT} - T_{IBTIM} - T_{ATIM\_WINDOW}) * I_{RX}$$

Transmitting Node: wake up, receive contention backoff, receive IBSS beacon, transmit #packets ATIM requests during ATIM window (receive contention, transmit ATIM), receive the remainder of ATIM window, send #packets (receive contention, transmit a transmit data packet), and return to back to sleep.

$$
\begin{aligned}
E'(t)_{Tn} = {} & (T_{CONT} * I_{RX}) + (T_{IBTIM} * I_{RX}) \\
& + [(T_{CONT} * I_{RX}) + (T_{ATIM} * I_{TX})] * PKTS \\
& + [T_{ATIM\_WINDOW} - (T_{CONT} - T_{ATIM}) * PKTS] * I_{RX} \\
& + [(T_{CONT} * I_{RX}) + (T_{DATA} * I_{TX})] * PKTS \\
& + [(T_{FRAME} - T_{LPM3} - T_{TnTransmissionSequence}) * I_{Sleep}] + E'_{LPM3}
\end{aligned}
$$

$$T_{TnTransmissionSequence} = T_{CONT} + T_{IBTIM} + T_{ATIM\_WINDOW} + (T_{CONT} + T_{DATA}) * PKTS$$

Receiving Node (All others): wake up, receive contention backoff, receive IBSS beacon, receive #packets ATIM requests during ATIM window (receive contention, receive ATIM packet), receive the remainder of ATIM window, receive #packets (receive contention and receive a data packet), and return to back to sleep.

$E'(t)_{Rn} = (T_{CONT} + T_{IBTIM}) * I_{RX}$
$+ (T_{CONT} + T_{ATIM}) * I_{RX} * PKTS$
$+ [T_{ATIM\_WINDOW} - (T_{CONT} - T_{ATIM}) * PKTS] * I_{RX}$
$+ (T_{CONT} + T_{DATA}) * I_{RX} * PKTS$
$+ [(T_{FRAME} - T_{LPM3} - T_{RnTransmissionSequence}) * I_{Sleep}] + E'_{LPM3}$

$T_{RnTransmissionSequence} = T_{CONT} + T_{IBTIM} + T_{ATIM\_WINDOW} + (T_{CONT} + T_{DATA}) * PKTS$

Entire Network: One "TBTT" AP node, #PKTS Transmitters, N - #PKTS – 1 Receivers

$E'(t)_{NetworkAve} = [\ E'(t)_{tbttAP} + (\#PKTS * E'(t)_{Tn}) + (N - \#PKTS - 1) * E'(t)_{Rn}\ ] / N$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

BMAC Model

T_{SAMP} = sample time

T_{POL} = preamble overlap

T_{INT} = network interval time = samp / dty

RInts = remaining intervals

T_{PRE} = preamble length = int * (pol + 1)

BMAC No Traffic Model

Nodes: Time is divided into n Intervals, each of which is equal to the (sample time / duty). For one interval, each node in the network will wake up and listen for sample time, after which it will go back to sleep for the remaining time in the interval.

$E'(t)_{NetworkAve} = (T_{SAMP} * I_{RX}) + (T_{INT} - T_{SAMP} - T_{LPM3}) * I_{SLEEP} + E'_{LPM3}$

$Lf(t)_{ave} = (Capacity * T_{Frame}) / E'(t)_{NetworkAve}$

BMAC Unicast & Regular Broadcast Traffic Model

Transmitting Node: Transmit #packets (transmit preamble, transmit Data Packet), sleep for the remaining intervals, except for sample time periods (listen for preambles sent by other nodes)

$E'(t)_{Tn} = [(T_{PRE} * I_{TX}) + (T_{DATA} * I_{TX})] * PKTS + [(T_{INT} - T_{SAMP} - T_{LPM3}) * I_{SLEEP} + E'_{LPM3}] * RInts$

Receiving Node: Receive #packets (receive 51% preamble, receive Data Packet), sleep for the remaining intervals, except for sample time periods (listen for preambles sent by other nodes)

$E'(t)_{Rn} = [(T_{PRE} * 0.51 * I_{RX}) + (T_{DATA} * I_{RX})] * PKTS + [(T_{INT} - T_{SAMP} - T_{LPM3}) * I_{SLEEP} + E'_{LPM3}] * RInts$

Entire Network: One Transmitter and (N – 1) Receivers.

$E'(t)_{NetworkAve} = [E'(t)_{Tn} + (N-1) * E'(t)_{Rn}]/ N$

$Lf(t)_{ave} = (Capacity * T_{INT}*(RInts +1)) / E'(t)_{NetworkAve}$

BMAC Rogue Denial of Sleep Broadcast Traffic Model

All Nodes: Receive #packets (receive 51% preamble, receive data packet), sleep for the remaining intervals, except for sample time periods (listen for preambles sent by other nodes).

$E'(t)_{NetworkAve} = E'(t)_{Rn}$
$= [(T_{PRE} * 0.51 * I_{RX}) + (T_{DATA} * I_{RX})] * PKTS + [(T_{INT} - T_{SAMP} - T_{LPM3}) * I_{SLEEP} + E'_{LPM3}] * RInts$

$Lf(t)_{ave} = (Capacity * T_{INT}*(RInts +1)) / E'(t)_{NetworkAve}$

ACKNOWLEDGMENT: These equations were developed by Michael Brownfield, Theresa Nelson, and Yatharth Gupta. Working in WSN independent study under the supervision of Michael Brownfield, Mike and Theresa wrote up the equations and Yatharth programmed them in MATLAB. Both Michael and Theresa developed MS Excel models which eventually verified the MATLAB models.

Vita

Lieutenant Colonel Michael I. Brownfield is an Information Systems Engineer in the United States Army Signal Corps. After receiving the Ph.D. in electrical engineering from Virginia Polytechnic Institute and State University, he will be assigned as an Assistant Professor in the Department of Electrical and Computer Science, United States Military Academy, West Point, NY. LTC Brownfield holds a BS degree in Electrical Engineering from the United States Military Academy and a MS degree in Electrical Engineering from Stanford University. He also earned a Professional Engineer (PE) certification from the Commonwealth of Virginia.

Following his graduate studies at Stanford University, Michael taught cadets as an Assistant Professor in Electrical Engineering at the United States Military Academy. In addition to his academic achievements, he has served in the U.S. Army since 1982 with duties serving from an enlisted radio repairman to a Company Commander in the 101st Airborne Division (Air Assault). He deployed from Europe as a Signal Brigade Automation Officer during Operation Desert Shield and Storm (Persian Gulf War). His current research interests are wireless sensor networks and mobile ad hoc networks.

Honor Societies:

Tau Beta Pi (Engineering, 2006)

Upsilon Pi Epsilon (Computer Science, 2002)

Eta Kappa Nu (Electrical Engineering, 2001)

Phi Kappa Phi (all academic disciplines, 1989)

Publications:

M. Brownfield, Y. Gupta, and N. Davis, "Wireless Sensor Network Denial of Sleep Attack," In *6th Annual IEEE Systems, Man, and Cybernetics (SMC) Information Assurance Workshop (IAW)*, pp. 356–364, June 2005.

M. Brownfield and N. Davis, "Symbiotic Highway Sensor Network," In *IEEE 62nd Vehicular Technology Conference (VTC)*, pp. 2701-2705, September 2005.

M. Brownfield, A. Fayez, and N. Davis, "Wireless Sensor Network Radio Power Management," In *OPNETWORK 2005*, August 2005.

M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis, "Wireless Sensor Network Energy-Adaptive MAC Protocol," *IEEE Consumer Communications and Networking Conference2006 (CCNC 2006),* Volume 2, pp. 778-782, January 2006.

M. Brownfield, A. Fayez, T. Nelson, and N. Davis, "Cross-layer Sensor Network Radio Power Management," In *IEEE Wireless Communications and Networking Conference (WCNC 2006)*, April 2006.