Open Access

Wireless Sensor Network Radio Power Management and Simulation Models

Michael I. Brownfield*[,a], Theresa Nelson[b], Scott Midkiff [b] and Nathaniel J. Davis IV[c]

[a]*Department of Electrical and Computer Engineering, United States Military Academy, West Point, NY 10996, USA*

[b]*Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA*

[c]*Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA*

Abstract: Wireless sensor networks (WSNs) create a new frontier in collecting and processing data from remote locations. The IEEE 802.15.4 wireless personal area network-low rate (WPAN-LR) 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 (or motes) 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 WPAN-LR-based sensor platform radios operates at a 250 kbps data rate and does not provide adequate time to completely power off the radio between the 128-byte constrained IEEE 802.15.4 transmissions. A new radio power management (RPM) algorithm presented in this paper exploits additional energy-saving opportunities introduced with the new generation of faster platform transceivers. The RPM algorithm optimizes radio sleep capabilities by transitioning nodes to intermediate power level states. Additionally, this experimental mote research also provides characterizations for the radio power levels, the mote platform state transition times, and the state transition energy costs of an IEEE 802.15.4 compliant sensor platform for improved accuracy in simulating WSN energy consumption.

Keywords: Wireless sensor network, power management, energy-efficiency, medium access control (MAC), simulation parameters.

1. INTRODUCTION

Sensor networks monitor phenomena as diverse as moisture, temperature, speed, and location using a wide variety of detectors. Since wireless sensor networks (WSNs) operate in a broadcast medium, these networks require a medium access control (MAC) layer to resolve contention in a random, multi-access environment. In efforts to make inexpensive sensors ubiquitous, these sensor platforms tend to have limited processor capability, memory capacity, and battery life. In dynamic ad hoc network environments, WSNs have the additional challenge of self-adapting to changes in topology, traffic loads, and existing battery conditions.

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 sensor platform transceivers. Short duration sleep offered by a network allocation vector (NAV) sleep mechanism provides significant energy savings [1-3]. 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 paper 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.

The remainder of this paper is organized as follows: the next section describes the WSN mote platform energy consumption model and WSN MAC protocol design in order to illustrate the opportunities for intermediate sleep levels, the parameters required to accurately model IEEE 802.15.4-compliant mote platforms, and the leading WSN MAC protocols used for RPM performance analysis. Section 3 explains the experimental circuit used to measure the platform state energy consumption and transition costs, the specific software and hardware component configurations to attain the intermediate LPM levels, and the RPM algorithm to attain significant energy savings by transitioning to these LPM levels. Section 4 describes the simulation models of two leading WSN MAC protocols in OPNET^{TM} Modeler and analyzes the enhanced network lifetime performance obtained by integrating the RPM algorithm. Finally, Section 5 summarizes the research with concluding remarks.

*Address correspondence to this author at the Department of Electrical and Computer Engineering, United States Military Academy, West Point, NY 10996, USA; E-mail: michael.brownfield@us.army.mil

Table 1. Receive and Sleep-Mode Current Consumption

Radio	Receive Mode	Power-Down Mode
CC2420 (Chipcon, 2006)	19.7 mA	0.1 µA
CC1000 (Chipcon, 2006)	9.6 mA	0.2 µA
RFM TR1001 (2005)	3.8 mA	0.7 µA

2. WSN ENERGY LOSS AND MAC PROTOCOL DESIGN

WSN MAC protocols extend network lifetimes by reducing the activity of the highest energy-demanding component of the sensor platform – the radio. Sacrificing network throughput and latency (delay), these protocols create opportunities for radios to sleep with active duty cycles reaching as low as 2.5% under minimal traffic conditions [4]. Understanding the sources of energy loss is essential in designing any power control system. Typical sources of energy loss in WSNs include idle listening, frame collisions, protocol overhead, and message overhearing.

Idle listening occurs when a station, or node in the WSN, listens to an inactive medium. This idle listening mode dominates power losses in networks characterized by scarce traffic and limited sleep cycles. 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. Table **1** illustrates how receiving a message consumes three to four orders of magnitude more energy than the radio power-down mode.

Frame collision occurs when a wireless sensor node sends a MAC frame, or message, which collides or overlaps in time with another message. If the interfering signal strength of the colliding packet sufficiently reduces the receiving node's signal-to-noise ratio (SNR), the interfering packet can corrupt the data at the receiving end. In single-channel radios, the radio cannot simultaneously receive while in the transmit mode. Therefore, the message sender's only indication of a collision is the absence of a message acknowledgement (ACK) from the receiver. The leading causes for wireless frame collisions are: 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. Resending messages causes both the sending and receiving node to expend additional energy.

Techniques employed by protocol designers to mitigate the effects of frame collisions include the IEEE 802.11 exponential contention backoff, transmission scheduling with TDMA protocols, and four-way request to send (RTS) – clear to send (CTS) – data –acknowledgement (ACK) handshaking to reserve the medium before sending data. Both the RTS and CTS messages contain a duration field which advertises to all surrounding nodes the length of the transmission exchange. By having all nodes set their network allocation vector (NAV) countdown timers for the duration of the exchange, a WSN protocol can significantly reduce frame collisions after the initial RTS-CTS exchange has seized the channel.

Protocol overhead in wireless networks 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. For example, adding data message headers and 2-to-1 Manchester encoding to the RF Monolithic TR1001 [5] transceiver reduces the 115.2 kbps physical transmission rate to an effective 46 kbps [4]. This 60% reduction does not yet include the additional network control overhead required to configure the network using neighbor discovery, synchronization time, and available message route determination. Protocol designers must minimize the protocol overhead while maintaining energy efficiency, latency and throughput requirements.

Message overhearing is receiving and discarding messages intended for other nodes and is tolerable in networks not constrained by energy. 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. In many WSN platforms, the radio receive mode actually expends more energy than the transmission. Message passing is an energy-efficient technique to reduce message overhearing using a four-way request to send (RTS) – clear to send (CTS) – data –acknowledgement (ACK) handshake to reserve the medium before sending data. Both the RTS and CTS messages contain a duration field which advertises to all surrounding nodes the length of the transmission exchange. Nodes set their network allocation vector (NAV) countdown timers for the duration of the exchange. Message passing provides a means for nodes to schedule a NAV sleep period after an overheard RTS-CTS handshake sequence by extracting the message duration field and scheduling a NAV table interrupt [6]. To reduce the probability of costly retransmissions and added latency, message passing also uses RTS-CTS exchanges to gain medium access and then transmits a burst of fragments of the larger message. As shown in Fig. (**1**), the receiver responds with an acknowledgement (ACK) message after each successful fragment transmission.

3. WSN SENSOR MAC PROTOCOLS

Sensor MAC (SMAC) is a WSN MAC protocol which represents the baseline energy-efficient protocol designed to extend WSN network lifetime [7]. SMAC divides a time frame into listen and sleep periods. 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 to form virtual clusters of nodes with the same active listen and sleep periods. By creating a small active duty cycle, node lifetimes can be significantly extended with bounded throughput and latency

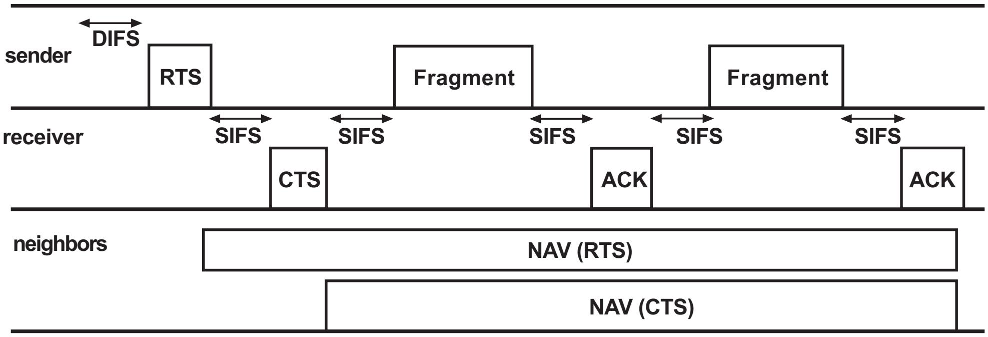

Fig. (1). Message passing timing and signaling (Ye *et al*., 2002).

tradeoffs. Sensors that border two synchronized clusters have the option of choosing one or both sleep schedules.

The bi-directional traffic in Fig. (**2**) represented by arrows illustrates how creating a slotted starting time for all network traffic and concentrating the traffic into a smaller active time frame reduces idle listening, trading off latency and throughput. To minimize collisions, nodes use the IEEE 802.11 standard exponential contention backoff for all channel access attempts. Furthermore, SMAC also reduces energy consumption using the message passing techniques employed for overhearing avoidance.

Timeout MAC (TMAC) is a WSN MAC protocol that decreases idle listening in WSN networks by establishing a dynamic sleep cycle [1]. TMAC nodes vary their active message exchange period depending on current traffic conditions. Unlike the SMAC static duty cycle, the TMAC dynamic duty cycle uses adaptive listening to attain significant energy savings and accommodate various network traffic loads experienced during a WSN's lifetime. TMAC nodes also form virtual clusters and automatically determine the initiation of a cluster sleep cycle based upon an adaptive timeout (TA) mechanism. To provide for multi-hop network communication, the TA period represents the worst case delay a CTS response packet could undergo before being transmitted. Equation (1) highlights the parameters used to calculate the TA period:

$$TA = 1.5 * (t_{SIFS} + t_{CWmax} + t_{RTS}) \quad (1)$$

where t_{SIFS} is the duration of a short interframe spacing, t_{CWmax} is the duration of the longest contention window backoff, and t_{RTS} is the duration of a RTS packet. Simulations indicated a need to scale this TA period by 50% for effective message exchange. Fig. (**2**) illustrates how TMAC effectively condenses the same amount of traffic as SMAC into a smaller dynamic time window to save energy by reducing idle listening at the expense of increased message delay.

4. RADIO POWER MANAGEMENT

The radio power management (RPM) algorithm creates graduated sleep modes for additional opportunities to transition the radio to lower power states. The mote platform energy and transition time 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.

4.1. 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. Fig. (**3**) shows that

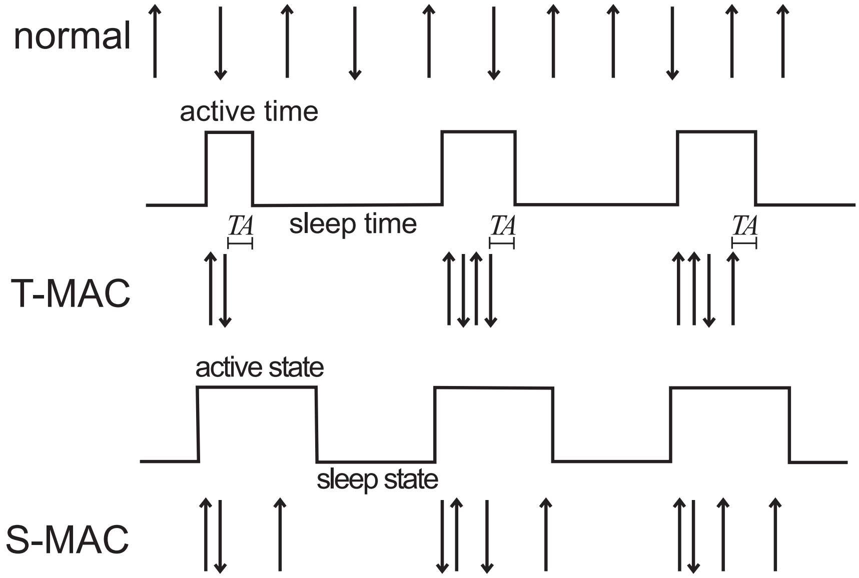

Fig. (2). SMAC static and TMAC dynamic duty cycles (vanDam and Langendoen, 2003).

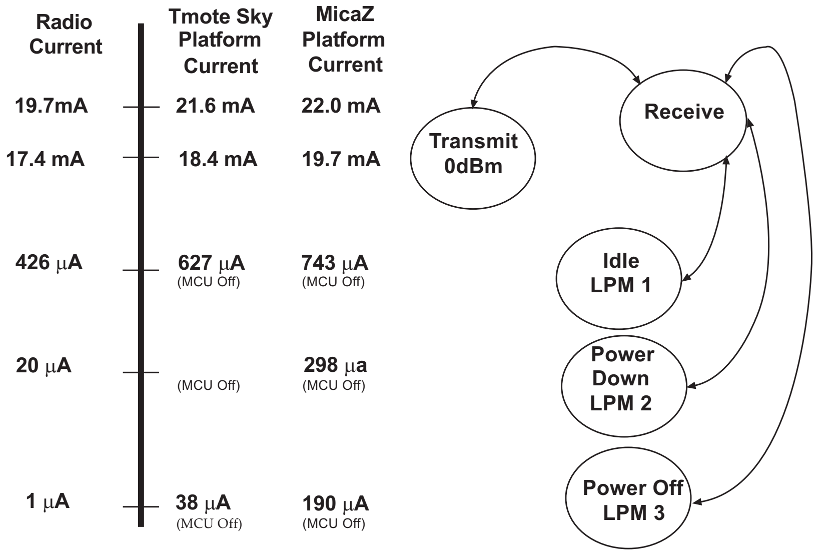

Fig. (3). CC2420 radio energy modes and platform energy allocations.

the Chipcon Corporation IEEE 802.15.4-compliant CC2420 radio [8] 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 Tmote sensor platform (21.6 mA) and a sleeping platform (38 µA) is 5.8 days vs. 9.01 years (or battery shelf life).

Sleep transition measurements of the CC2420 radio integrated onto the Tmote platform indicate a 6.81 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 448 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 Fig. (**3**) show the static radio mode platform energy costs in terms of current consumption, 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 [4]. The experimental current (I) measurements indicate that the average transition cost of 1.88 mA for a Tmote receive-LPM3-receive transition is two orders of magnitude larger than the average LPM3 sleep mode base current (38 µA) and an order of magnitude smaller than the receive mode current (21.56 mA). Therefore, transition energy costs will have a impact on network lifetime. Additionally, the time required to recover from the LPM3 mode (6.81 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.

4.2. 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 Fig. (**4**)

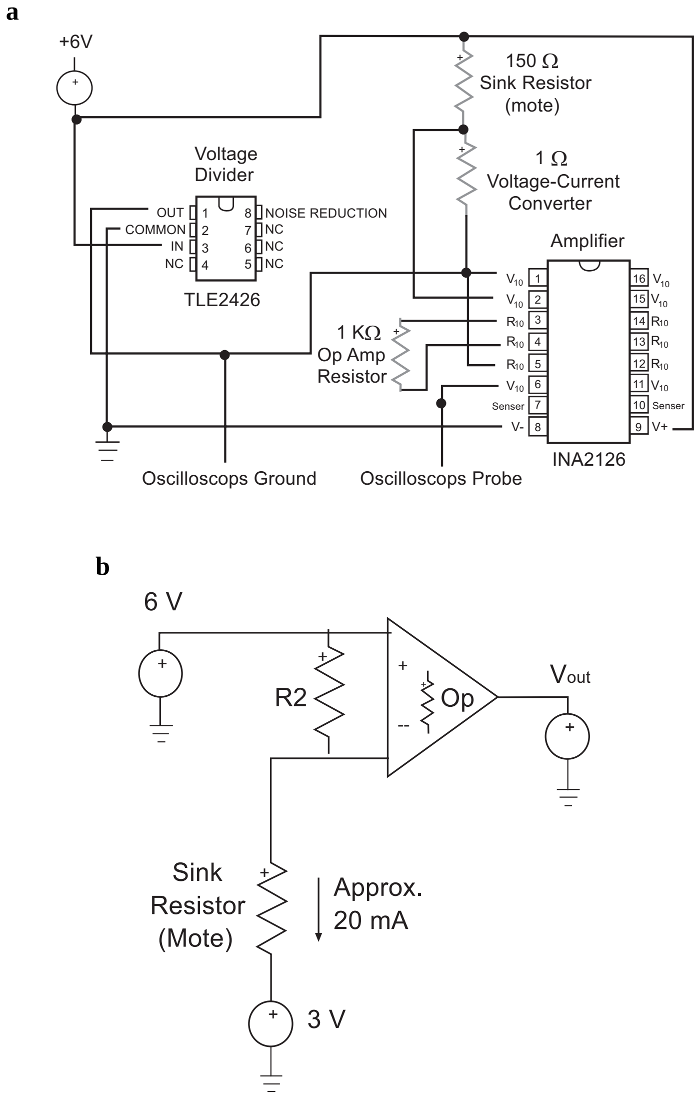

Fig. (4). Instrumentation circuit (top) and equivalent amplification circuit (bottom).

details the connections to amplify the sensor platform's current flowing through the 1-ohm R2 resistor [9]. A fixed voltage source was used to provide consistent regulated power throughout the experiment. The right side of Fig. (**4**) illustrates a block diagram of the instrumentation circuit and the method used to calibrate the amplification circuit. To calibrate the circuit, a Rmote resistor was chosen to produce a typical mote 20 mA current flow through the amplifier's input circuit across R2. 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. Once the circuit gain was established, the sink resistor was replaced with the battery terminals of the mote platform to provide regulated power to the mote and measure the current consumption and transition times.

4.3. Mote Platform Radio Low Power Mode Descriptions

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 [10] 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. Next, the microcontroller reset the voltage regulator (VReg) on the CC2420 radio. The crystal oscillator

Table 2. 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				

(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 which indicate packet reception notification (FIFOP) and PHY preamble completion (start frame delimiter). With all hardware subcomponents now enabled, the radio was in the receive mode and prepared for incoming messages. The component configurations for the transition to sleep for each LPM level followed a reverse sequence of events as shown in Table **2**. The oscilloscope screen captures in Fig. (**5**) illustrate each of the various stages of the platform power up for the Tmote and MICAz platforms.

The experimental data for the energy costs and transition delays for each of the three LPM sleep levels are presented in Table **3**. 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 Fig. (**6a-c**).

4.4. 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 [11] and Crossbow MICAz [12] 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 Fig. (**7**) with many 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 [2]. 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 **3** establish the RPM transition thresholds, Table **4** illustrates the potential energy savings regained using LPM1 and LPM2 for the various packet data sizes and their associated RTS durations. The IEEE 802.15.4-compliant 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.

5. OPNET RPM SIMULATION MODEL

A simulation scenario was designed in OPNET™ Modeler to compare the energy efficiency of the SMAC and TMAC. The simulation scenario was composed of 20 WSN nodes operating with a 500 ms active/sleep frame period, and

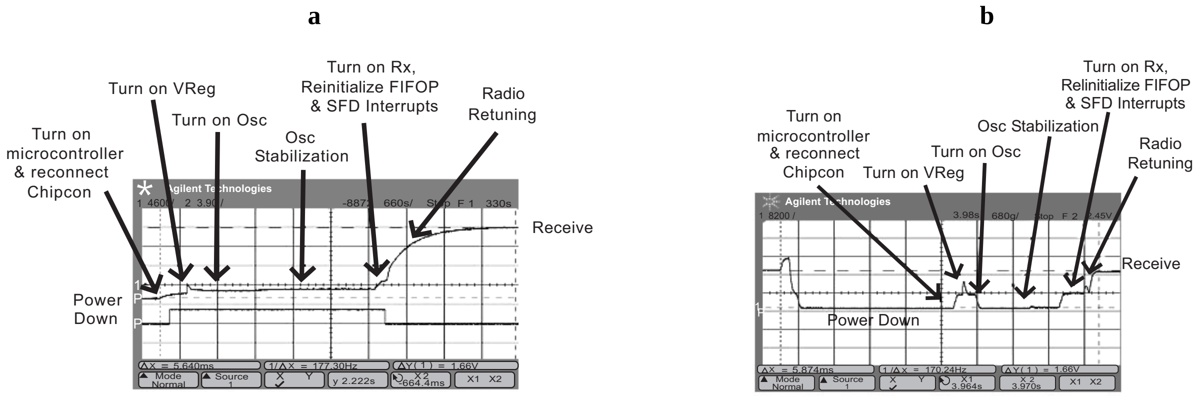

Fig. (5). LPM 3 transition power up current consumption: tmote (left) and MICAz (right).

Table 3. 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

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 **3**: 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.

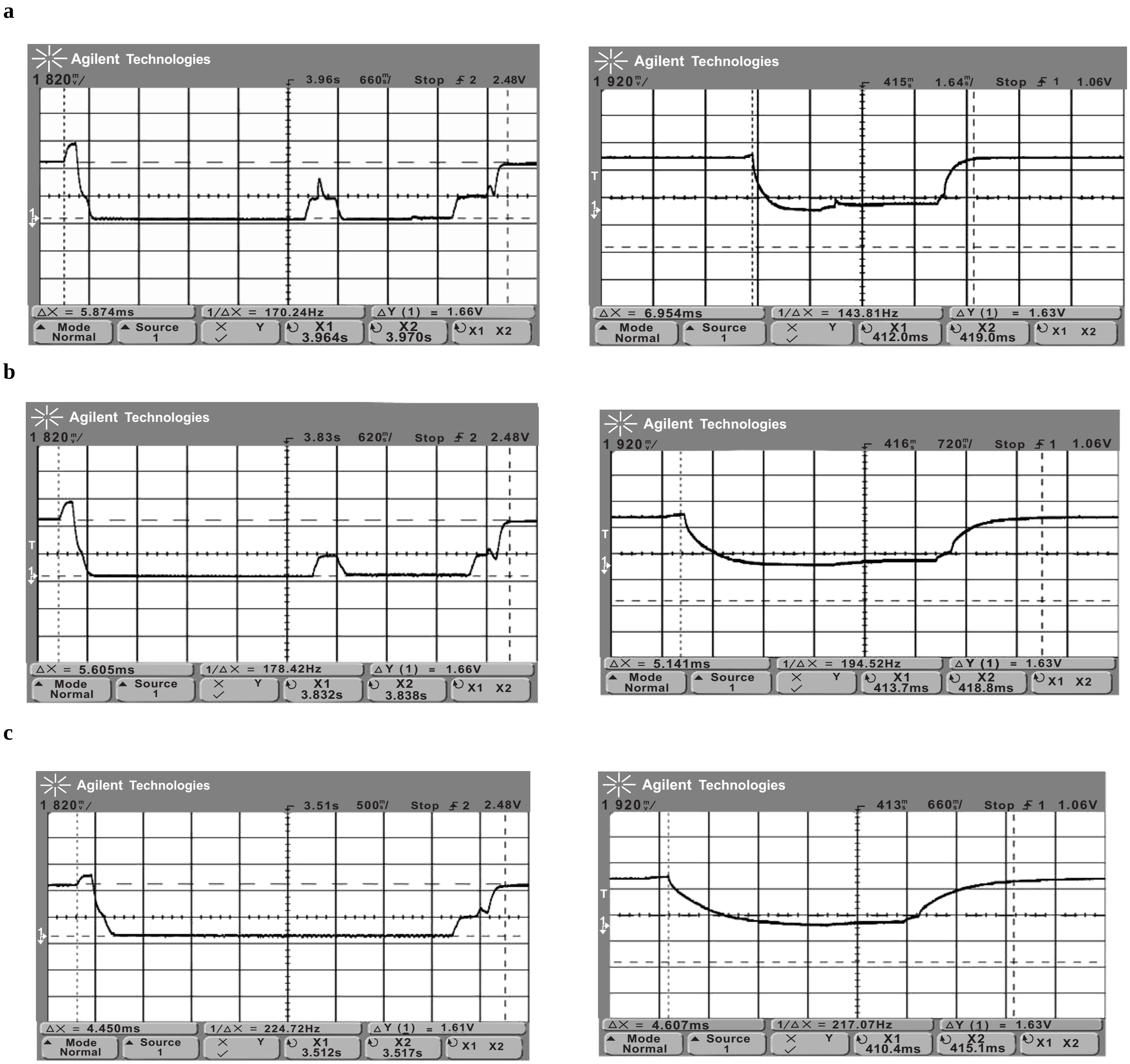

Fig. (6). a Tmote LPM3 (left) and MicaZ LPM3 (right) current levels.
b Tmote LPM2 (left) and MicaZ LPM2 (right) current levels.
c Tmote LPM1 (left) and MicaZ LPM1 (right) current levels.

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 **4**, 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 Figs. (**8-11**), 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.

The SMAC model simulations in Figs. (**8** and **9**) show that the message passing NAV sleep method was unable to save any appreciable energy over the No-NAV sleep model. As previously shown in table 3, the radio draws more current during the receive than the transmit mode due to the low noise amplifier (LNA) in the receiver and low transmit

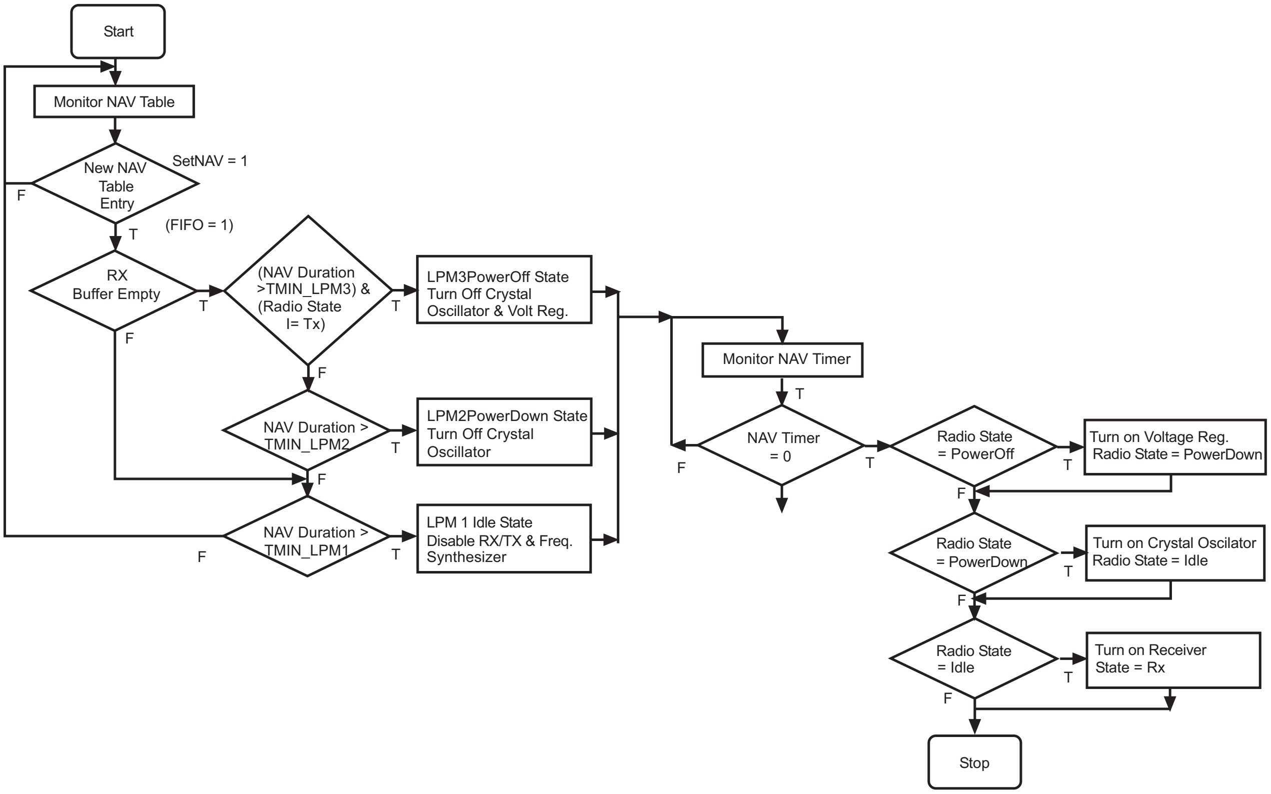
Start
Monitor NAV Table
New NAV Table Entry
SetNAV = 1
F
T
(FIFO = 1)
RX Buffer Empty
T
F
(NAV Duration >TMIN_LPM3) & (Radio State != Tx)
T
F
LPM3PowerOff State Turn Off Crystal Oscillator & Volt Reg.
NAV Duration > TMIN_LPM2
T
F
LPM2PowerDown State Turn Off Crystal Oscillator
NAV Duration > TMIN_LPM1
T
F
LPM 1 Idle State Disable RX/TX & Freq. Synthesizer
Monitor NAV Timer
T
NAV Timer = 0
F
T
Radio State = PowerOff
T
F
Turn on Voltage Reg. Radio State = PowerDown
Radio State = PowerDown
T
F
Turn on Crystal Oscillator Radio State = Idle
Radio State = Idle
T
F
Turn on Receiver State = Rx
Stop

Fig. (7). Radio power management algorithm.

Table 4. **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.

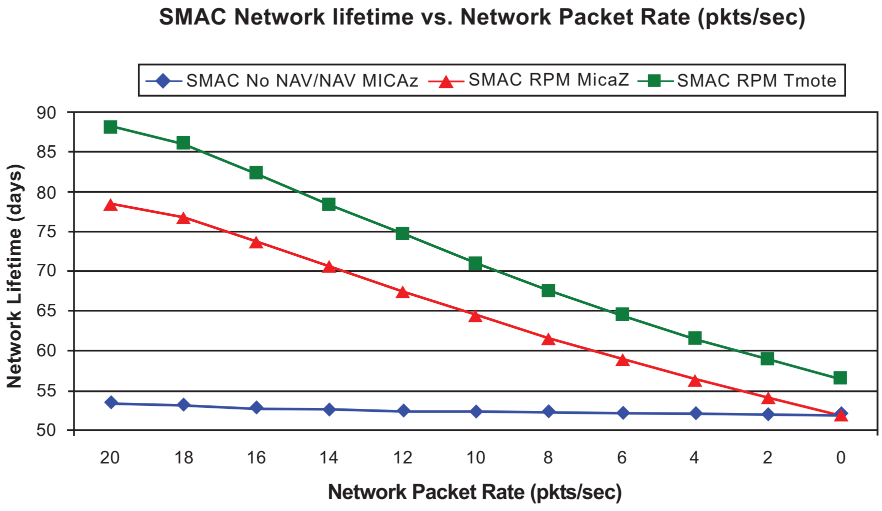

Fig. (8). SMAC WSN traffic network lifetime performance.

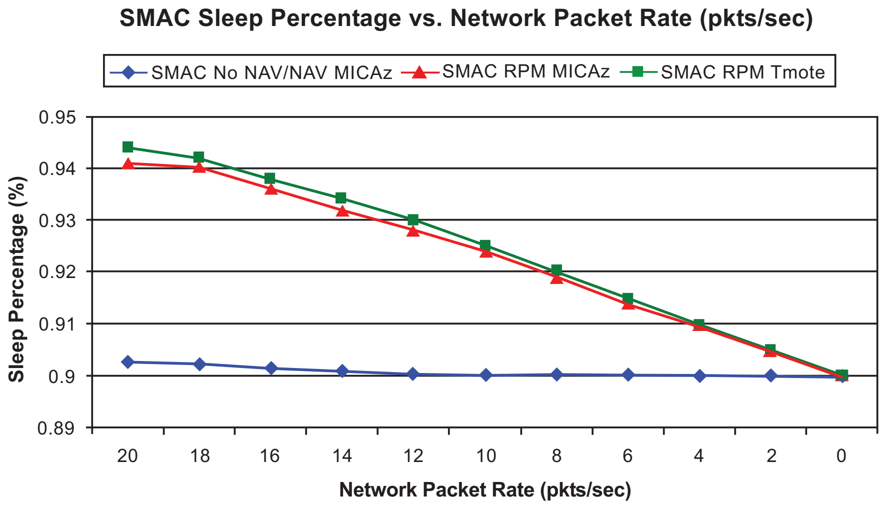

Fig. (9). SMAC WSN average sleep percentage.

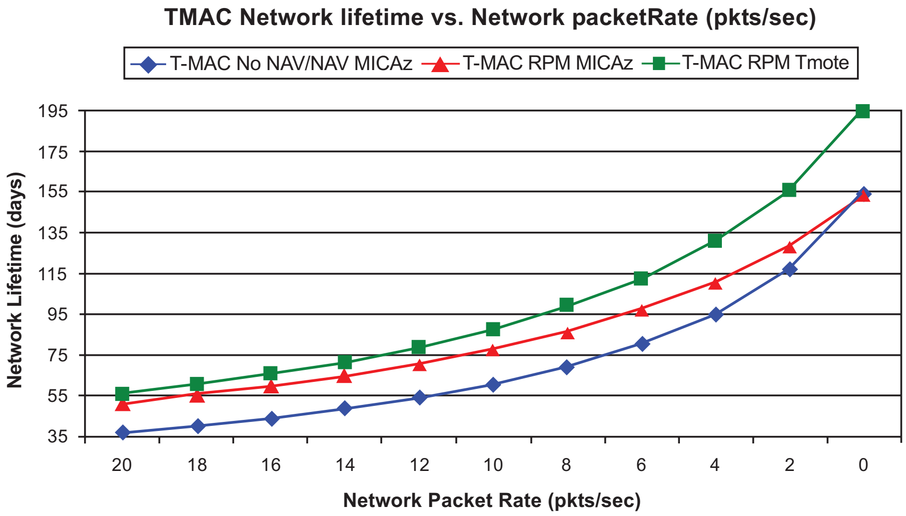

Fig. (10). TMAC WSN traffic network lifetime performance.

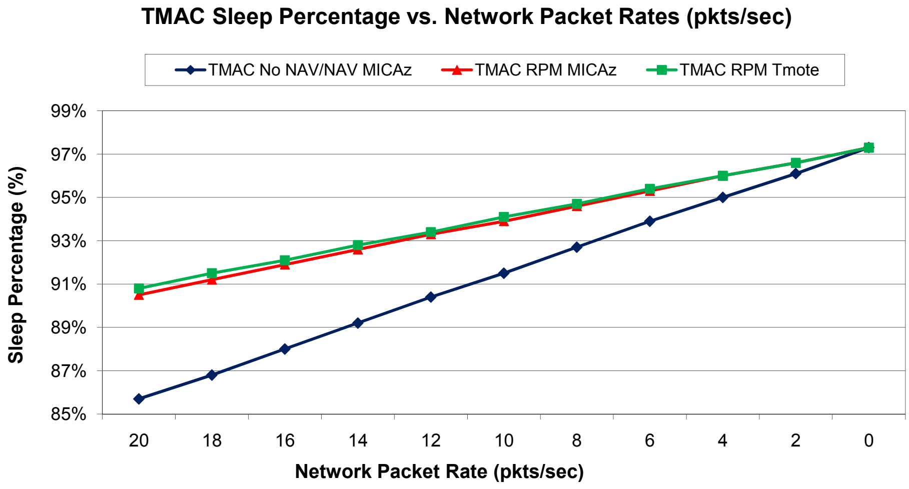

Fig. (11). TMAC WSN average sleep percentage.

power (1dBm) of this class of radio; therefore, the higher packet rates for the NO-NAV sleep model used less power and increased the network lifetime. 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 **4**, 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. Fig. (**10**) 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.

6. CONCLUSIONS

This paper 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 vital tools for researchers to continue their progress with the next generation of wireless sensor network platforms.

REFERENCES

[1] T. van Dam, and K. Langendoan, “Energy-efficient MAC: An adaptive energy-efficient MAC protocol for wireless sensor networks”, In: *Proc IEEE Int Conf Embed Networked Sensor Syst (Sensys 2003)*, Los Angeles, CA, November 5-7, 2003, pp. 171-180.

[2] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks”, In: *Proc IEEE/ACM Networking Trans*, June 2004, Vol. 12, No. 3, pp. 493-506.

[3] M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis, “Wireless sensor network energy-adaptive MAC protocol,” In: *IEEE Consumer Commun Networking Conf 2006 (CCNC 2006)*, January 2006, Vol. 2, pp. 778-782.

[4] K. Langendoen, and G. Halkes, *Energy Efficient Medium Access Control*, Delft University of Technology, Netherlands, Technical Paper , 2004.

[5] RF Monolithics, “*TR1001 868.35 MHz Hybrid Transceiver*”, http://www.rfm.com [Accessed April 2005].

[6] S. Sing, and C. Raghavendra, “PAMAS: Power Aware multi-access protocol with signaling for ad hoc networks,” In: *ACM Comp Commun Review*, July 1998, Vol. 28, No. 3, pp. 5-26.

[7] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” In: *Proc Joint IEEE Joint Conf IEEE Comp Commun Soc (Infocom 2002)*, June 23-27, 2002, pp. 1167-1576.

[8] Chipcon Corporation, CC2420 Data Sheet [Online], http://www.chipcon.com [Accessed February 2006].

[9] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” In: *Proc IEEE Int Conf Embed Networked Sensor Syst (Sensys 2004)*, November 2004, pp. 95-107.

[10] TinyOS, http://webs.cs.berkeley.edu/tos [Accessed June 2005].

[11] MoteIV Corporation, Tmote Sky, http://www.moteiv.com, [Accessed February 2006].

[12] CrossBow Corporation, MICAZ Series Sensor Processor / Radio Module. http://www.xbow.com [Accessed February 2006].

Received: February 13, 2009 Revised: January 11, 2010 Accepted: January 15, 2010