Effects of Denial-of-Sleep Attacks on Wireless Sensor Network MAC Protocols

David R. Raymond, *Member, IEEE*, Randy C. Marchany, *Associate Member, IEEE*, Michael I. Brownfield, *Senior Member, IEEE*, and Scott F. Midkiff, *Senior Member, IEEE*

***Abstract*—Wireless platforms are becoming less expensive and more powerful, enabling the promise of widespread use for everything from health monitoring to military sensing. Like other networks, sensor networks are vulnerable to malicious attack. However, the hardware simplicity of these devices makes defense mechanisms designed for traditional networks infeasible. This paper explores the *denial-of-sleep* attack, in which a sensor node's power supply is targeted. Attacks of this type can reduce the sensor lifetime from years to days and have a devastating impact on a sensor network. This paper classifies sensor network denial-of-sleep attacks in terms of an attacker's knowledge of the medium access control (MAC) layer protocol and ability to bypass authentication and encryption protocols. Attacks from each classification are then modeled to show the impacts on four sensor network MAC protocols, i.e., Sensor MAC (S-MAC), Timeout MAC (T-MAC), Berkeley MAC (B-MAC), and Gateway MAC (G-MAC). Implementations of selected attacks on S-MAC, T-MAC, and B-MAC are described and analyzed in detail to validate their effectiveness and analyze their efficiency. Our analysis shows that the most efficient attack on S-MAC can keep a cluster of nodes awake 100% of the time by an attacker that sleeps 99% of the time. Attacks on T-MAC can keep victims awake 100% of the time while the attacker sleeps 92% of the time. A framework for preventing denial-of-sleep attacks in sensor networks is also introduced. With full protocol knowledge and an ability to penetrate link-layer encryption, all wireless sensor network MAC protocols are susceptible to a full domination attack, which reduces the network lifetime to the minimum possible by maximizing the power consumption of the nodes' radio subsystem. Even without the ability to penetrate encryption, subtle attacks can be launched, which reduce the network lifetime by orders of magnitude. If sensor networks are to meet current expectations, they must be robust in the face of network attacks to include denial-of-sleep.**

***Index Terms*—Medium access control (MAC), wireless security, wireless sensor networks (WSNs).**

I. Introduction

Wireless sensor networks (WSNs) are becoming increasingly attractive for a variety of application areas, including industrial automation, security, weather analysis, and a broad range of military scenarios. The challenge of designing these systems to be robust in the face of myriad security threats is a priority issue. One such threat is the *denial-of-sleep* attack, which is a specific type of denial-of-service (DoS) attack that targets a battery-powered device's power supply in an effort to exhaust this constrained resource. If a large percentage of network nodes, or a few critical nodes, are attacked this way, the network lifetime can be reduced from months or years to days.

The impacts of denial-of-sleep attacks on WSN MAC protocols have not been thoroughly addressed. The only previous study that focused on denial-of-sleep in WSN is [1], which models the network lifetime under routine traffic patterns for a representative set of MAC protocols and the impact of a *denial-of-sleep broadcast attack* on these protocols on the Tmote Sky [2] WSN platform. This paper describes a more potent unauthenticated broadcast attack in which a back-to-back stream of unauthenticated packets is transmitted, as opposed to the attack used in [1], which uses a much lower rate of four attack packets per second. This paper also explores the impacts of constant physical-layer jamming, intelligent replay, and a full domination attack for each of the protocols considered. We also expand on [1] by modeling the impact of these attacks on the Crossbow Mica2 [3] WSN platform in addition to Tmote Sky. Furthermore, the impacts of various denial-of-sleep attacks on current wireless sensor devices are validated through implementation on the Mica2. A framework for defending against these potentially devastating attacks is then presented.

To make the nodes small and inexpensive for economical deployment in large numbers, they generally have very limited processing capability and memory capacity. Because the design of these devices usually favors decreased cost over increased capabilities, we cannot expect Moore's law to lead to enhanced performance. Another challenge unique to sensor node platforms is their extremely limited and often nonreplenishable power supply. Mica2 and Tmote Sky are two examples of widely available sensor node platforms. Both devices are configured to run for a year or more on a pair of AA batteries, relying on long periods of sleep to save power. The dominant source of power loss in these sensor platforms is the radio subsystem. Table I shows the instantaneous power consumption during receive, transmit, and sleep periods for these devices [2], [3]. The data link layer, specifically the medium access control

Manuscript received June 4, 2007; revised October 28, 2007, December 20, 2007, and March 5, 2008. First published March 31, 2008; current version published January 16, 2009. The work of S. F. Midkiff was supported in part by the National Science Foundation. Any opinion, finding, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the United States Army. The review of this paper was coordinated by Prof. M. Guizani.

D. R. Raymond is with the United States Army's Battle Command Training Program, Fort Leavenworth, KS 66027 USA (e-mail: raymondd@vt.edu).

R. C. Marchany is with the IT Security Laboratory, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060 USA (e-mail: marchany@vt.edu).

M. I. Brownfield is with the United States Military Academy, West Point, NY 10996-1905 USA (e-mail: michael.brownfield@usma.edu).

S. F. Midkiff is with the Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0111 USA (e-mail: midkiff@vt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2008.921621

TABLE I
SENSOR PLATFORM POWER CONSUMPTION AND SLEEP TRANSITION DATA

		Mica2	Tmote Sky
Power Draw	Receive (mW)	36.81	64.68
	Transmit (mW)	87.90	55.20
	Sleep (mW)	0.090	0.114
	Radio Platform	CC1000	CC2420
Sleep Transition Time (ms)		2.60	6.81
Maximum Data Rate (kbps)		76.8	250

(MAC) protocol, is responsible for managing the radio. Therefore, the MAC protocol must keep the radio in a low-power sleep mode as much as possible. As a result, most research in the area of sensor node power conservation is focused on MAC protocols.

The MAC protocols considered in this paper include the slotted carrier sense multiple access with collision avoidance (CSMA/CA) protocols Sensor MAC (S-MAC) [4], Timeout MAC (T-MAC) [5], and Berkeley MAC (B-MAC) [6]. In addition, Gateway MAC (G-MAC) [7] is also considered here, which is a clustered protocol that combines a contention-based slot reservation period with a time-division multiple-access (TDMA) period for data dissemination. Similar centralized cluster-based WSN protocols include low-energy adaptive clustering hierarchy (LEACH) [8] and power-aware clustered TDMA (PACT) [9].

The rest of this paper is organized as follows. Section II explores sources of energy loss in sensor networks and briefly describes S-MAC, T-MAC, B-MAC, and G-MAC. Section III discusses related work in the area of sensor network security. Section IV outlines a framework for classifying denial-of-sleep attacks in these networks, and Section V explores the impact of a selection of denial-of-sleep attacks against the MAC protocols presented in Section II. Section VI validates the effectiveness of the attacks presented by describing their implementation on the Mica2 wireless sensor platform and goes on to analyze the efficiency of these attacks. Finally, Section VII provides a framework for defending against denial-of-sleep attacks in sensor networks, and Section VIII offers conclusions.

II. SENSOR NETWORK MAC PROTOCOLS

MAC layer protocols designed for WSNs use various algorithms to save battery power, e.g., by placing the radio in low-power modes when not actively sending or receiving data. Table I illustrates the importance of maximizing a node's sleep ratio because the transmit and receive power can be up to three orders of magnitude greater than the sleep power. Let the sleep ratio R_{sleep} be equal to $T_{\text{sleep}}/(T_{\text{active}} + T_{\text{sleep}})$, where T_{active} and T_{sleep} are the active and sleep times, respectively. A node's lifetime is expressed as

$$T_{\text{sensorlife}} = \frac{C_{\text{battery(mWh)}}}{(R_{\text{sleep}})(P_{\text{sleep(mW)}}) + (1 - R_{\text{sleep}})(P_{\text{active(mW)}})} \quad (1)$$

where P_{active} and P_{sleep} are the active and sleep mode power draws, respectively, and C_{battery} is the total amount of available energy. P_{active} is almost three orders of magnitude greater than P_{sleep}, so it is important to keep nodes in sleep mode as much as possible. For example, Tmote Sky consumes 64.68 mW in receive mode and 0.114 mW in sleep mode [2]. Using two standard 3000-mAh AA batteries, it will last 3300 days in sleep mode, but only 5.8 days in receive mode. The disparity between receive and sleep costs leads to an exponential increase in network lifetime as the sleep time increases, suggesting that an attack that decreases the sleep time by even a few percentage points can have a dramatic impact on network lifetime.

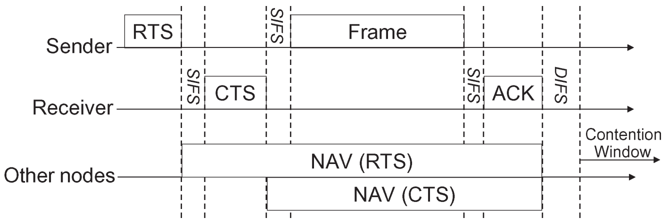

Fig. 1. Typical NAV scenario. SIFS is the wireless protocol's SIFS. DIFS is the distributed interframe space. These interframe delays are used to coordinate access to the wireless medium.

A. Sources of Energy Loss

The amount of power that can be saved largely depends on the MAC protocol's ability to overcome the radio's four primary sources of energy loss, i.e., *collisions*, *control packet overhead*, *overhearing*, and *idle listening*.

1) Collisions: Collision loss refers to the energy wasted due to packet collisions on the wireless medium. If a transmission of sufficient signal strength interferes with a data packet being sent, the data will be corrupted at the receiving end. Corrupted data can sometimes be recovered using error-correcting codes (ECCs); however, ECCs add transmission overhead, which is contrary to the goal of reducing the radio transmit time.

2) Control Packet Overhead: Depending on the MAC protocol used, control packets may have to be received by all nodes within radio range of the sender, resulting in power drain in a potentially large number of nodes. If nodes can be forced to stay awake for spurious control packets, the battery life can be greatly impacted. Examples of control packets are the *request-to-send (RTS)* and *clear-to-send (CTS)* messages used by the IEEE 802.11 protocols.

3) Overhearing: Overhearing loss refers to the energy wasted by a node having its radio in receive mode while a packet is being transmitted to another node. Most WSN MAC protocols reduce overhearing by trying to ensure that a node is only awake when there is traffic destined for it. One way to prevent overhearing is to ignore packets destined for other nodes after hearing an RTS/CTS exchange. After overhearing RTS and CTS, nodes set a network allocation vector (NAV) interrupt based on the message duration field in the CTS message and then go to sleep [10]. The NAV represents the duration of the entire RTS/CTS/Data/ACK sequence. Fig. 1 depicts a typical NAV scenario.

Opportunities for NAV sleep are significantly reduced on new platforms because the time required to transition to sleep and back is longer than the packet transmit times for even the longest packets. Tmote Sky, for example, takes 6.81 ms to transition the radio from receive to sleep mode and back [11],

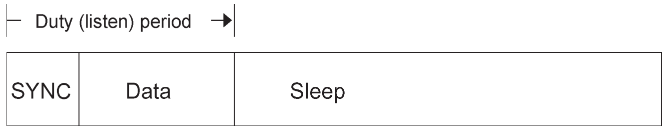

Fig. 2. S-MAC frame structure.

whereas the time required to send a maximum-sized IEEE 802.15.4-compliant frame of 128 B is only 4.09 ms.

4) Idle Listening: A node's radio consumes the same amount of power simply monitoring the channel as it does when it is receiving data. If a node can be made to listen even when there is no traffic destined for it, power is wasted.

B. WSN MAC Protocols

Section V analyzes the impact of denial-of-sleep attacks against S-MAC, T-MAC, B-MAC, and G-MAC. Selecting a particular protocol for a given sensor network deployment depends on several factors such as the sensor platform selected, the energy efficiency of the protocol, the expected duration of the deployment, the expected amount of unicast and broadcast traffic, and the memory footprint required by the protocol.

1) S-MAC: The S-MAC protocol [4] uses a fixed duty cycle, with a default of 10%, during which, traffic is exchanged between nodes. Radios in networks using this protocol will be asleep 90% of the time, thereby producing an almost tenfold improvement in node life. In S-MAC, sensor nodes organize themselves into virtual clusters using periodic broadcast synchronization (SYNC) messages. Upon deployment, a node will listen for a SYNC message. If it does not hear one before timeout, it will broadcast a SYNC message announcing its sleep cycle. Nearby nodes overhear this message and synchronize their schedules to the sending node. SYNC messages are periodically repeated to the correct time drift and keep the virtual clusters' sleep cycles synchronized. If a node overhears two SYNC messages, it will adapt both duty cycles to maintain network connectivity. Fig. 2 depicts the S-MAC frame structure.

2) T-MAC: T-MAC [5] improves on S-MAC by concentrating all traffic at the beginning of the duty period, as depicted in Fig. 3, thus trading network latency for power conservation. The arrows in the figure indicate transmitted and received messages. T-MAC uses the same SYNC mechanism to form virtual clusters as S-MAC. Instead of remaining awake for a set period, however, an adaptive timeout (TA) mechanism allows nodes to transition to sleep mode when there is no more traffic in the cluster. According to [5], TA is set based on the longest time that a hidden node would have to wait before hearing the beginning of a CTS response message as

$$TA = 1.5 \times (C + R + T) \tag{2}$$

where C is the length of the contention interval, R is the time to send an RTS packet, and T is the time between the end of an RTS packet and the beginning of a CTS packet, which is the duration of a short interframe space (SIFS). The TinyOS implementation of T-MAC for the Mica2 platform uses a slightly different calculation of TA, which is discussed in Section VI. The improvement in network lifetime using this protocol depends on the amount of traffic in the network. In [5], T-MAC is shown to have up to a fivefold increase in network lifetime over S-MAC.

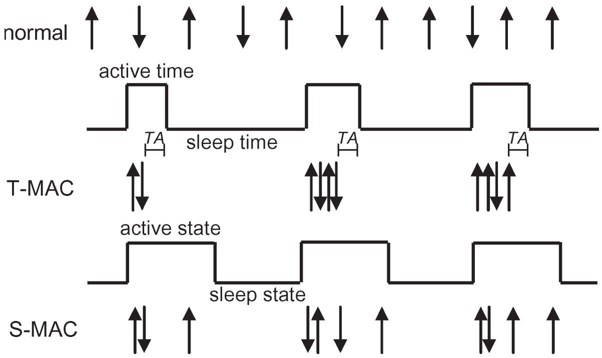

Fig. 3. T-MAC adaptive timeout.

3) B-MAC: B-MAC [6] does not form clusters of nodes or attempt to synchronize sleep schedules. Instead, it uses a technique called *low-power listening* (LPL) to reduce energy consumption. In LPL, nodes briefly awaken at a fixed interval and check the wireless channel for valid preamble bytes that indicate a pending data transmission from another node. A node with data to send transmits a preamble that is longer than the interval between receiver samplings to ensure that all nearby nodes have the opportunity to detect the preamble and receive the subsequent data packet. The interval between channel sensings, or the *check interval*, is set based on average network node degree and traffic levels [6]. Fig. 4 depicts the sending and receiving node behavior in B-MAC. Polastre *et al.* showed that under ideal conditions, B-MAC could have duty cycles as low as 1% in a low-traffic network [6].

4) G-MAC: G-MAC [7] is an energy-efficient MAC protocol designed to coordinate transmissions within a cluster. Fig. 5 depicts the G-MAC frame structure, which is divided into a collection period and a contention-free distribution period. During the collection period, nodes that have outgoing unicast or broadcast traffic transmit a future RTS (FRTS) message to a gateway node. Traffic destined for other clusters is also transmitted to the gateway node during the contention period using an RTS/CTS/DATA/ACK exchange. At the end of the contention period, the cluster head, or gateway, transmits a gateway traffic indication message (GTIM) that provides a mechanism for cluster synchronization while broadcasting a schedule of message transactions between nodes. Nodes then exchange data during the contention-free period. The gateway is elected using a periodic resource-adaptive election process in which nodes volunteer based on current resource levels. New elections are indicated by a flag in the GTIM message. G-MAC eliminates overhearing, except for a minimum amount of control traffic that a node might overhear while waiting to transmit an FRTS during the contention period.

III. Current Research in Sensor Network Security

Most research on sensor network security focuses on integrity and confidentiality. This section first introduces basic WSN security mechanisms and then reviews recent research on DoS in sensor networks.

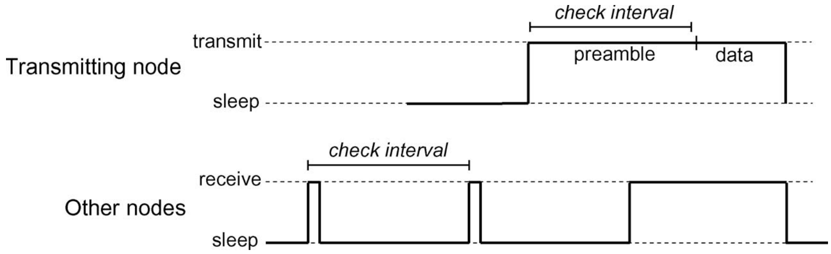

Fig. 4. B-MAC sender and receiver behavior.

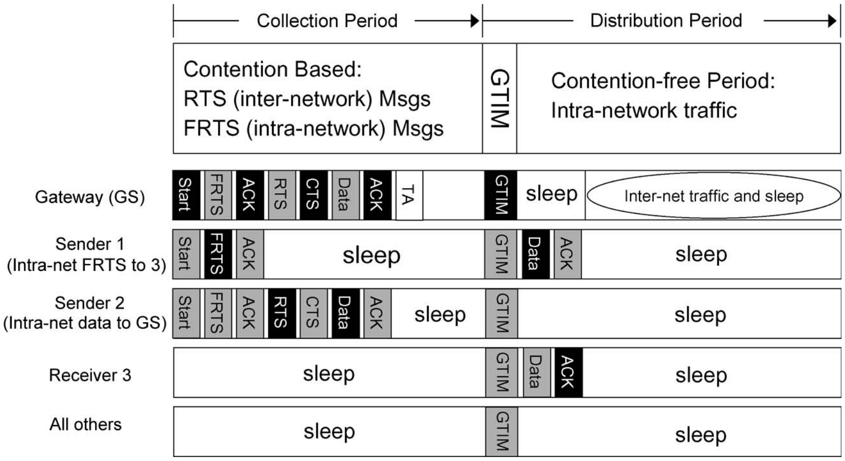

Fig. 5. G-MAC frame structure.

A. General Sensor Network Security Mechanisms

Security Protocols for Sensor Networks (SPINS) [12] was one of the first attempts to provide security services for resource-constrained WSN platforms. The SPINS suite was designed to provide data confidentiality and authentication using symmetric encryption. SPINS also provides a mechanism to support data freshness for unicast transmissions using monotonically increasing counter values shared by sender and receiver. These counter values are the encryption nonce values used for encryption using the counter (CTR) block cipher mode of operation. Since the counter is maintained by both sender and receiver, it does not need to be transmitted with outgoing packets, thus reducing the transmission overhead. If a packet is properly decrypted, the receiving node knows that the sender's counter matches its counter, so it is not a replayed packet. However, a dropped packet interrupts the sequence numbers and requires an expensive challenge-response handshake to resynchronize counters. Furthermore, this anti-replay mechanism requires every node to maintain a table of counter values listing every node from which it receives a packet, and each node must share a secret key with every communication partner. The memory requirements for storing such information make it unrealistic in memory-constrained sensor nodes, even in a moderately sized network of 25 nodes [13].

TinySec is a link-layer security architecture specifically designed for sensor networks [13]. It provides support for simple authentication and authenticated encryption, where a *message authentication code* is calculated over an encrypted message. The energy consumption overhead in TinySec is relatively low and is primarily caused by longer transmission times due to increased packet length. The per-packet power consumption is increased by 10% for authenticated encryption and 3% for authentication only. While TinySec provides a good architecture for data confidentiality and integrity, it does not prevent message replay. The authors also recognize the threat of *resource consumption* attacks; however, defense against such attacks is outside the scope of their work [13].

The IEEE 802.15.4 specification [14], also known as *ZigBee*, details the architectural requirements for a particular class of wireless radios and protocols for personal area network devices and wireless sensor nodes. The specification provides hardware support for data confidentiality and integrity in compliant devices, mandating the use of *Advanced Encryption Standard* (AES) encryption and message integrity codes (MICs) to provide support for access control, data encryption, and frame integrity. Support for defense against replay attack, in the form of frame counters, is optional according to the standard.

B. DoS in WSNs

Physical-layer jamming can simultaneously prevent traffic flow on a WSN and rapidly drain sensor batteries. A potential defense against power consumption caused by jamming is to go to a low-power state when such attacks are in progress, waking only periodically to sense the channel [15]. A prerequisite for such a mechanism is for nodes to identify that a jamming attack is ongoing. In [16], jamming attacks are classified as either *constant*, *deceptive*, *random*, or *reactive*. Constant jamming normally involves a constant high-power transmission that requires maximum energy by an attacker and may not be feasible if the attacker is under similar power constraints as the target network. A *deceptive jammer* sends a constant stream of packets into the network to make it appear that the medium is being filled with legitimate traffic. A *random jammer* randomly alternates between sleep and jamming to save power. Finally, a *reactive jammer* only sends a jam signal when it senses traffic to cause collisions.

Techniques for identifying the jamming attacks explored in [16] include statistical analysis of received signal strength indicator (RSSI) values, average time required to sense an idle channel (*carrier sense time*), and packet delivery ratio (PDR). All of these techniques require that the network not be jammed upon deployment so that baseline measurements can be taken and none of them alone identify all types of jamming. By combining techniques and introducing the notion of a consistency check, however, all four types of jamming can reliably be identified. One such algorithm first identifies poor link utility through PDR analysis and then uses a statistical RSSI analysis as a consistency check to determine whether the poor network performance is due to jamming. A second technique compares PDR values with those expected based on the location of neighbor nodes as a consistency check, assuming neighbor locations are known.

The *denial-of-sleep broadcast attack* is presented in [1], where the impact of a malicious host obeying the MAC layer protocol and broadcasting unauthenticated traffic into the network is modeled. Although the malicious broadcast traffic is dropped due to authentication failure, network lifetime is significantly reduced for networks using the S-MAC and T-MAC protocols. The authors introduce the G-MAC protocol, which weathers this type of malicious broadcast attack particularly well. In G-MAC, requests to broadcast traffic must be authenticated by the gateway node before the traffic can be sent to other nodes; therefore, only the gateway suffers power loss due to unauthenticated broadcast.

In a *replay attack*, network traffic is recorded and replayed. The replayed data will be treated just as it was the first time it was sent over the network, and it will be received by a subset of nodes. If there is no replay protection, the traffic will be accepted as legitimate and forwarded to the destination, thus consuming resources on each node along the path.

IV. Classifying MAC Layer Denial-of-Sleep Attacks in WSNs

The MAC-layer denial-of-sleep attacks on WSNs can be categorized based on the level of protocol knowledge required to initiate them and the level of network penetration achieved by an attacker. Penetration refers to an attacker's ability to read and send trusted traffic. A network is easily penetrated if the networking protocols are known and if cryptographic mechanisms are not used for communication or are compromised. While there are mechanisms available for secure communication in WSN, they are not as robust as those found in traditional networks due to resource constraints. Any shared medium can be attacked with physical-layer jamming. Jamming, however, is a blunt instrument for executing a denial-of-sleep attack on WSN. Depending on the MAC protocol, the lifetime of a WSN can be significant, even in the face of jamming, requiring that an attacker jam the network for an extended period to render it ineffective. Furthermore, conducting a jamming attack requires considerable resources. A more efficient attack strategy is to use knowledge of MAC protocols to initiate an assault aimed at draining power from sensor platforms, thereby rendering the network unusable and nullifying any other security mechanisms. In the ensuing discussion, the following three classifications of MAC layer denial-of-sleep attacks are used.

1) Class 1—No Protocol Knowledge, No Ability to Penetrate Network: With no knowledge of the MAC layer protocols, attacks are limited to physical-layer jamming and unintelligent replay attacks. In an *unintelligent replay attack*, recorded traffic is replayed into the network, causing nodes to waste energy receiving and processing these extra packets. If nodes in the network do not implement an anti-replay mechanism, this attack causes the replayed traffic to be forwarded through the network, consuming power at each node on the path to the destination. Undetected replay has the added benefit (to the attacker) of causing the network to resend data that could subvert the network's purpose. For example, replaying traffic in a military sensor network deployed to sense enemy movement could cause combat units to be misdirected.

2) Class 2—Full Protocol Knowledge, No Ability to Penetrate Network: Traffic analysis can determine which MAC protocol is being used in a sensor network. With this knowledge, an attacker could expand the attack types beyond those listed earlier to include *intelligent jamming*, injecting unauthenticated unicast or broadcast traffic into the network, or being more selective about replaying previous traffic. Intelligent jamming [17] uses knowledge of link-layer protocols to reduce network throughput without relying on a constant jam signal, for example, by jamming only RTS packets. Such attacks improve over constant physical-layer jamming in that they preserve attacker energy, which can be important if attacking nodes have constraints similar to those of the target nodes. Even when attacker power consumption is not a factor, intelligent jamming might be used to make it more difficult for a network to detect an attack.

If valid source and destination addresses are inserted by an attacker, unauthenticated traffic requires that nodes stay awake to receive packets, even if they are later discarded due to invalid authentication. If packets are encrypted, a node must receive the entire packet before decrypting and discarding it. The number of nodes impacted by unauthenticated broadcast traffic depends on the MAC protocol. For example, if the protocol uses a cluster head or gateway node to authenticate broadcast traffic before other nodes are compelled to receive it, then only the gateway node energy is impacted. Replay attacks can also be more cleverly executed with knowledge of the protocol, even if the messages cannot be deciphered. It has been shown that if the MAC layer protocol is known, traffic analysis can be used to distinguish data from control traffic [18]. Depending on the protocol, effective denial-of-sleep attacks can be mounted by replaying specific control messages, even without the ability to decrypt the traffic. For example, properly timed SYNC retransmission in the S-MAC protocol could potentially prevent nodes from entering their duty/sleep cycle and could keep all nodes in receive mode until their batteries are depleted.

TABLE II
CLASSIFICATION OF WSN DENIAL-OF-SLEEP ATTACKS

	Class I	*Class II*	*Class III*
Attack Type	*No protocol knowledge, no network penetration*	*Full protocol knowledge, no network penetration*	*Full protocol knowledge, network penetrated*
Constant jammer	√	√	√
Deceptive jammer	√	√	√
Random or reactive jammer	√	√	√
Intelligent jammer		√	√
Untrusted unicast/broadcast		√	√
Trusted rogue unicast/bcast			√
Unintelligent replay	√	√	√
Intelligent replay		√	√
Full domination			√

3) Class 3—Full Protocol Knowledge, Network Penetrated: Attacks in this category could be devastating to a WSN. With full knowledge of the MAC protocol and the ability to send trusted traffic, an attacker can produce traffic to gain maximum possible impact from denial-of-sleep attacks. The types of attacks that could be executed against each MAC protocol and the impact of such attacks are analyzed in Section V.

Table II classifies the types of denial-of-sleep attacks available based on the attacker's protocol knowledge and ability to penetrate the network. A fourth case, i.e., no knowledge of the protocol but an ability to penetrate the network, is not considered since the ability to penetrate the network assumes full knowledge of the MAC layer protocol.

V. EFFECTS OF DENIAL-OF-SLEEP ATTACKS ON SELECTED MAC PROTOCOLS

In this section, attacks from each of the three classifications and their impacts on S-MAC, T-MAC, B-MAC, and G-MAC are analyzed. This section explores the impacts of constant physical-layer jamming, unauthorized broadcast, intelligent replay, and a full domination attack for each of the three protocols considered. A full domination attack assumes that the attacker has penetrated the network and has full knowledge of the MAC protocol. In each case, a full domination attack can reduce the network lifetime to ten days for the Mica2 platform and six days for the Tmote Sky platform, which is equivalent to a network lifetime under IEEE 802.11 with no power saving features.

A. Network Model

Each network is modeled in MATLAB using similar configurations. The Mica2 models are based on the TinyOS protocol implementations available on Sourceforge.net [19]. Since none of these protocols have been implemented for CC2420-based platforms at the time of this writing, the Tmote Sky models assume the basic functionality of the protocols and are adapted to the increased data rate of the CC2420 transceiver and the specified IEEE 802.15.4 interframe spacing duration. Table III provides network and protocol parameters for the Mica2 and Tmote Sky networks. G-MAC is specifically designed for ZigBee-compliant platforms and is, therefore, not included in the Mica2 models. The B-MAC check interval of 20 ms for Tmote Sky was determined to give the longest network lifetimes for our model. The LPL sampling cost for CC2420 is based on the cost of transitioning from power-down mode to receive mode and back [11].

TABLE III
NETWORK AND PROTOCOL ANALYTICAL MODEL PARAMETERS

	Mica2	Tmote Sky
Number of nodes	50	50
Effective data rate	19.2 kbps	250 kbps
Frame duration	1300 ms	500 ms
Legitimate traffic rate	1 pkt per frame	2 pkts per frame
Payload size	29 bytes	64 bytes
S-MAC duty cycle	10%	10%
T-MAC sleep timeout (TA)	81 ms	13.5 ms
B-MAC check interval	100 ms	20 ms
G-MAC GTIM size	NA	20 bytes
G-MAC collection period	NA	250 ms
G-MAC distribution period	NA	250 ms

The system models 50 Crossbow Mica2 or Tmote Sky nodes in a single-hop neighborhood. Network lifetimes are based on Mica2 and Tmote Sky power consumption for receive, transmit, and sleep given in Table I. IEEE 802.11 with no power management provides the baseline case. In regular IEEE 802.11, the radio is always in receive mode unless transmitting, resulting in a ten-day lifetime for the Mica2 network and a six-day lifetime for the Tmote Sky network. All traffic is node to node, and the effects of all denial-of-sleep attacks are regardless of legitimate traffic rates.

B. Denial-of-Sleep Attacks and Impacts

The results of each of the attacks are given in Table IV. In our models, transmit and receive pairs for all traffic are randomly assigned in a uniform distribution to equally distribute energy consumption across the nodes. We assume that all nodes are simultaneously deployed with fresh batteries and that new nodes are not added to the network during its lifetime. Network lifetime is defined as the average time between network deployment and the time that nodes' power supplies are exhausted.

1) Physical-Layer Jamming Attack: The first attack classification in Section IV considers an attacker with no protocol knowledge and no ability to penetrate the network. This classification of attack is modeled using a deceptive jamming attack, as described in [16], in which a constant stream of bytes is broadcast into the network. Under this attack, S-MAC is unable to transmit data and nodes remain awake during the entire 10% duty cycle because they are not able to enter NAV sleep. T-MAC fares much worse under this type of attack because nodes must sense an idle channel for the period dictated by the network's adaptive sleep timeout (TA) before going to sleep. Under deceptive jamming, nodes will never sense an idle channel and will continuously delay their sleep cycle, thus constantly remaining in receive mode. This results in a network lifetime of ten days on Mica2 and six days on Tmote Sky. B-MAC also suffers under deceptive jamming. A constant stream of preamble bytes forces all nodes in the network into a cycle of receiving and analyzing preamble bytes looking for the beginning of the data transmission. Therefore, nodes remain in receive mode during the entire jamming period,

TABLE IV
EFFECTS OF DENIAL-OF-SLEEP ON NETWORK LIFETIME ON TMOTE SKY AND MICA2 FOR SELECTED MAC PROTOCOLS (NETWORK LIFETIME GIVEN IN DAYS)

Platform	Routine Network Traffic				Attack Traffic (see Table II for classifications)			
					Class I	*Class II*		*Class III*
	MAC Protocol	Empty Network	Unicast Traffic	Broadcast Traffic	Deceptive Jamming	DoS Broadcast	Intelligent Replay	Full Domination
Mica2	802.11	10	10	10	10	10	10	10
	S-MAC	99	122	98	99	99	10	10
	T-MAC	227	119	160	10	10	10	10
	B-MAC	775	140	140	10	18	18	10
Tmote Sky	802.11	6	6	6	6	6	6	6
	S-MAC	56	56	56	56	56	6	6
	T-MAC	194	111	133	6	6	6	6
	B-MAC	120	58	58	6	10	10	6
	G-MAC	1024	828	295	237	371	160	6

resulting in a lifetime of ten days on Mica2 and six days on Tmote Sky. In the G-MAC protocol, the gateway node will constantly remain awake because there is no network idle time to allow it to go to sleep and will, therefore, last for six days on Tmote Sky. Other nodes will wake up and timeout once during each frame listening for a GTIM. Waking up for these small GTIM messages results in 0.16% duty cycle and a lifetime of 1287 days (or battery shelf life) for all the other nodes in the network. A more effective attack against G-MAC would be to periodically lift the jamming attack so that a new gateway is elected, thereby distributing maximum power draw among all the nodes. This would cause the average node per frame power consumption to be

$$P_{\text{NodeAverage}} = \frac{(P_{\text{Gateway}}) + (n-1)(P_{\text{OtherNodes}})}{n} \tag{3}$$

where n is the number of nodes, P_{Gateway} is the gateway power consumption while always awake, and $P_{\text{OtherNodes}}$ is the power consumption of the rest of the nodes. Under this attack, the G-MAC network lifetime is reduced to 237 days.

2) DoS Unauthenticated Broadcast Attack: The second attack classification considers an attacker with full protocol knowledge but no ability to penetrate the network. In this case, the attacker broadcasts traffic into the network following all the MAC protocol rules for timing and collision avoidance. Under S-MAC, T-MAC, and B-MAC, these messages are received by all nodes, but are discarded because they cannot be authenticated. Although the broadcast messages are not authenticated, the fact that all nodes stay awake to receive the messages has a significant impact on network lifetime. Sensor nodes using the S-MAC protocol are unable to save power using NAV sleep, keeping them in receive mode during their entire 10% duty cycle and resulting in a network lifetime of 99 days on Mica2 and 56 days on Tmote Sky. To minimize network lifetime for networks running the T-MAC protocol, short broadcast messages are sent at a period just short of the adaptive timeout (TA) to prevent nodes from going to sleep. This attack will keep the sensor nodes awake during the entire frame and reduce lifetime to ten and six days for Mica2 and Tmote Sky, respectively, while keeping the attacker's power requirements to a minimum. Since B-MAC nodes do not synchronize schedules, the most effective broadcast attack against B-MAC is to transmit back-to-back packets. Nodes must remain awake to overhear, on average, one half of the preamble plus the duration of the data packet. The percentage of time that a Mica2 is awake per broadcast packet is determined as follows, where T_{preamble} and T_{pkt} are the preamble duration and data packet duration, respectively:

$$\begin{aligned} \text{awake percentage} &= \left[\frac{\left(\frac{T_{\text{preamble}}}{2} + T_{\text{pkt}}\right)}{(T_{\text{preamble}} + T_{\text{pkt}})}\right] \\ &= \left[\frac{\left(\frac{113\text{ ms}}{2} + 17\text{ ms}\right)}{(113\text{ ms} + 17\text{ ms})}\right] \\ &= 56.5\%. \end{aligned} \tag{4}$$

The preamble duration of 113 ms and the packet duration of 17 ms are based on a 271-B preamble and 36-B data packet (with headers) transmitted at 19.2 kb/s. If the packets are broadcasted back to back, the overall awake percentage is the same as the per-packet awake percentage, allowing the victim to sleep 43.5% of the time. Substituting the preamble duration and data packet duration for Tmote Sky yields an awake percentage of 55.3%. This results in a network lifetime of 18 days for Mica2 and ten days for Tmote Sky.

Under G-MAC, only the gateway receives the broadcasted FRTS during the collection period. Since it cannot be authenticated, the broadcast message is not scheduled during the distribution period. To maximize the impact of this attack on G-MAC, the gateway should be kept awake during the entire collection period. The G-MAC gateway uses the same adaptive timeout mechanism as T-MAC to go to sleep during the contention period if there is no more traffic for it. An attacker should, therefore, send short broadcast messages at a rate just short of the adaptive timeout period to prevent the gateway from transitioning to sleep mode. Assuming no other traffic in the network, the other nodes would only wake up to receive an empty GTIM and, then, sleep for the remainder of the time, resulting in an overall network lifetime of 371 days on Tmote Sky. Any legitimate network traffic in addition to the unauthenticated broadcast packets further reduces this lifetime.

3) Intelligent Replay Attack: Another attack in the category of full protocol knowledge but no network penetration is an intelligent replay attack. If an attacker can distinguish control traffic from data traffic under S-MAC, SYNC packets can be replayed at an interval short of the sensor cluster's duty cycle, effectively restarting the duty cycle and pushing back the sleep period each time. This would keep all nodes awake until they run out of power. In G-MAC, FRTS messages should be replayed such that the corresponding NAV periods fill the contention-free portion of each frame. For a message size of 64 B, 75 FRTSs would fill the contention-free period, ensuring that at least one node is awake at all times. This effect, combined with a longer GTIM message that all nodes must receive, results in a network lifetime of 160 days, assuming all the FRTSs are for unicast packets. If any of the replayed FRTS messages happen to be broadcast FRTSs, the network lifetime is further degraded because all nodes must wake up during the contention-free period to listen for the broadcasts. If only 10% of the FRTS messages, or seven FRTS messages per frame, are for broadcast, the network life is cut by almost 50%, dropping to 83 days. The worst case is if all FRTSs are for broadcast messages. In this case, the network lifetime is reduced to 12 days as discussed hereinafter. Even if the message size is not known, the attacker could simply attempt to resend recorded FRTS messages until the gateway quits accepting them. The maximum number of FRTSs that an attacker can send can be determined based on the length of the collection period as follows:

$$N_{\mathrm{FRTS}} = \left(\frac{T_{\mathrm{CollectionPeriod}}}{T_{\mathrm{Cont}} + T_{\mathrm{DIFS}} + T_{\mathrm{FRTS}} + T_{\mathrm{SIFS}} + T_{\mathrm{ACK}}} \right) \qquad (5)$$

where T_{Cont} is the average contention period, T_{DIFS} and T_{SIFS} are the IEEE 802.11 distributed and short interframe space periods, T_{FRTS} is the time required to send a 13-B FRTS message, and T_{ACK} is the time required for the gateway to send a 5-B acknowledgment. With a 250-ms collection period, a maximum of 138 FRTS messages can be sent. With the potential for 138 FRTSs, the attacker can easily maximize traffic during the contention-free period.

4) Full Domination Attack: The final attack classification is one in which an attacker has full protocol knowledge and has penetrated the network. This type of attack might be mounted using one or more compromised nodes in the network. Once this level of network penetration is achieved, all of the MAC protocols are susceptible to worst-case power consumption. An attack against S-MAC is simply to send a SYNC message at a frequency just short of the duty cycle to keep delaying the transition to sleep mode. The T-MAC network lifetime is minimized by continually sending packets at an interval slightly shorter than the adaptive timeout (TA) so that none of the nodes can ever transition to sleep. Although not efficient for the attacker, a deceptive jamming attack is the most effective attack against B-MAC. A full domination attack against G-MAC has the attacker broadcasting a GTIM message before the gateway node by waiting for less than the required Point Coordination Function Interframe Space (PIFS) backoff normally required before a GTIM. If the attacker fills the GTIM with broadcast messages that fill the entire frame up to the next GTIM, all nodes will remain in receive mode during the entire frame waiting for the broadcast traffic. By repeating this pattern for each frame, all nodes are kept awake, and the network lifetime is reduced to six days on Tmote Sky. A simpler full-domination attack against G-MAC would simply have the attacker send broadcast FRTSs to the gateway such that the contention-free period is filled with broadcast messages. With eighty-nine 64-B packets, the 250-ms contention-free period would be filled, resulting in a 50% duty cycle for all nodes and a network lifetime of 12 days.

C. Discussion

The analysis of these attacks shows that with knowledge of the MAC protocol, even without the ability to penetrate encryption, attacks that have more significant impact on the network than constant physical-layer jamming can be constructed. Some attacks not only significantly reduce the network lifetime but also are subtle enough that the network may not even be able to identify that it is under attack. Furthermore, these attacks can be sustained longer because the attacker can conserve power by not transmitting a constant jam signal. The efficiency of these attacks is explored in the next section.

VI. Denial-of-Sleep Attack Implementations

Selected attacks were implemented and tested on the Crossbow Mica2 wireless sensor platform to validate the threat of denial-of-sleep attacks on S-MAC, T-MAC, and B-MAC described in Section V. The attacks were programmed in `nesc`, the programming language for TinyOS [20]. Attacks on G-MAC were not tested because the G-MAC protocol has not yet been implemented in the hardware. This section describes the details of these attacks and analyzes the efficiency with which they can be executed. An attack that minimizes power consumption for the attacker is preferred. If the attacker uses less power than the victim nodes, attacks can be executed using mote-class devices, which are easier to deploy and are less prone to detection because their physical-layer properties are similar to those of the deployed network devices. It is expected that by following the carrier-sense-based collision avoidance mechanisms inherent in these protocols, many of these denial-of-sleep attacks can be carried out in such a way that the targeted networks maintain throughput and latency similar to that of the network when it is not under attack. This analysis is a topic for future research.

The Mica2 platform was selected for these tests because it is a widely used commercially available sensor node platform that is compatible with the TinyOS operating system and with the S-MAC, T-MAC, and B-MAC implementations that were included in the TinyOS CVS release available on Sourceforge.net [19] when these tests were developed.[1] All experiments were carried out in a laboratory setting, with the nodes approximately 1.5 m apart in an open area. S-MAC and T-MAC parameters were used in their default settings, as downloaded from Sourceforge, except where otherwise specified. Those parameters are discussed in the following sections

[1] All tests with S-MAC, T-MAC, and B-MAC were performed with the implementations in `tinyos − 1.x/contrib/S − MAC/`, `tinyos − 1.x/contrib/T − MAC`, and `tinyos − 1.x/tos/platform/mica2` in the TinyOS CVS repository [19] as of August 1, 2006.

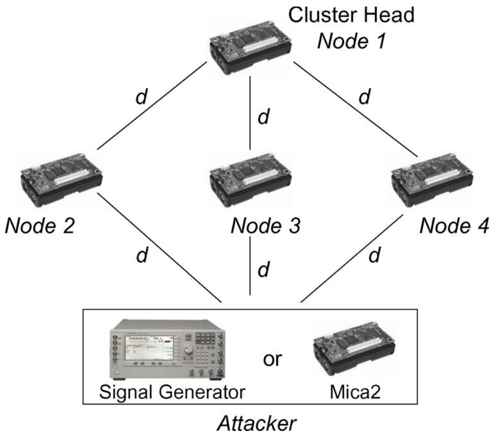

Fig. 6. Experimental setup for denial-of-sleep attacks ($d \simeq 1.5$ m).

where appropriate. B-MAC was configured using LPL-mode 3, which has a 135-ms check interval and uses 371-B preambles. In the S-MAC and T-MAC experiments, nodes were turned on and allowed to synchronize by exchanging SYNC packets before attacks were initiated. B-MAC nodes were turned on and allowed to enter LPL-mode before attacks were begun. Because these tests were not designed to measure the effects of attacks on network performance metrics such as throughput or latency, there was no traffic in the network other than MAC-layer SYNC packets used by S-MAC and T-MAC to synchronize schedules. Fig. 6 depicts the experimental layout. For the SYNC attacks described below, the cluster head node depicted in Fig. 6 was first started so that its schedule would be adopted by the other nodes in the one-hop neighborhood. A signal generator was used for the constant physical-layer jamming attacks against all protocols. A Mica2 sensor device identical to the victim nodes was used for all the other attacks.

The following paragraphs first describe and analyze attacks on networks using the S-MAC protocol corresponding to the three attack classes described in Section IV. They go on to analyze attacks on T-MAC and B-MAC. Finally, the efficiency with which these attacks can be executed is examined.

A. Attacks on S-MAC

1) Class 1—Constant Jamming Attack: Two types of constant jamming attacks were conducted to determine the impact of each on the S-MAC protocol. The first attack was executed using a signal generator to transmit a constant 10-dBm signal on the same frequency as the targeted sensor devices. The jam signal was fine tuned to the frequency of the sensor devices using a spectrum analyzer. Jamming was initiated after the targeted devices synchronized schedules. As expected, constant jamming did not impact the devices' ability to maintain their default 10% duty cycle, so nodes were not prevented from sleeping 90% of the time. The second type of attack was the deceptive jamming attack described in Section III-B. This attack was executed using a Mica2 device to transmit a constant stream of TinyOS preamble bytes (0xAA) after the victim nodes synchronized their schedules. This attack was also unsuccessful in keeping the victim nodes awake beyond their default duty cycle. These tests show that while an attacker is able to prevent communication between nodes using both constant jamming and deceptive jamming, it is not able to impact the lifetime of an S-MAC network. Furthermore, these attacks are inefficient because the attacker is awake and constantly transmitting while the victim nodes maintain their fixed duty cycle.

2) Class 2—Intelligent Replay Attack: As described in Section II-B, S-MAC uses periodic SYNC packets to prevent clock drift from desynchronizing the clustered nodes' schedules. Fig. 2 depicts the S-MAC frame structure. Each frame is divided into the SYNC period (during which SYNC messages are sent), a data period (during which broadcast data traffic is transmitted and unicast exchanges are initiated), and a sleep period (when nodes that are not involved in a unicast exchange place their radios in sleep mode to conserve energy). Frame length is determined at a compile time based on the control packet size, transceiver bandwidth, and duration of platform-specific DIFS and SIFS duration. The frame length for Mica2 using the default S-MAC configuration is 1300 ms. The default SYNC period is 47 ms and the data period is 83 ms for a 130-ms active period during each frame. The SYNC period at the beginning of each frame is the time set aside for sending SYNC packets; however, a receiving node processes SYNC packets whenever they are received. The default *SYNC_PERIOD*, or time between SYNC packets, is 12 000 ms or ten frames.

Each 10-B SYNC packet includes a 2-B *sleepTime*, which indicates to nodes receiving the packet when the sending node will next enter the sleep mode. When a node receives a SYNC packet from another node on its same schedule, it recalculates its next sleep time to maintain synchronization. Instead of simply resetting its next sleep time according to the value in the SYNC packet, the receiving node splits the difference between its next sleep time and the time in the received SYNC packet as follows:

$$sleepTime = \left\lfloor \frac{sleepTime + receivedSYNCpkt.sleepTime}{2} \right\rfloor. \quad (6)$$

This allows nodes on the same schedule to improve synchronization over time rather than having them radically change their schedules upon receipt of a SYNC message.

A replay attack is executed by recording and replaying SYNC packets. Even if they are encrypted, these packets are easily identified by an attacker monitoring a network cluster by their size and timing. S-MAC SYNC packets are 10-B long and occur during the first few milliseconds of an S-MAC frame. WSN encryption mechanisms are careful to minimize overhead to limit the added data transmission overhead, which consumes unnecessary power. Therefore, even if all packets are encrypted, various types of packets are still identifiable because their sizes increase by a constant amount.

Sending a constant stream of back-to-back SYNC packets is sufficient to keep targeted nodes awake. To maximize attack efficiency, an attacker should send SYNC packets as far apart as possible to minimize the attacker awake time while still achieving the desired effect of keeping the targeted nodes awake.

Recall that each node receiving a SYNC packet calculates its new sleep time according to (6). The correct interval for a SYNC replay attack is, therefore, slightly less than 1/2 of the *sleepTime* specified in the SYNC packet. The receiving node's next sleep will be scheduled for approximately $\lfloor(1.5 \times sleepTime)/2\rfloor$, before which, the node will receive another SYNC packet delaying its next sleep opportunity by the same amount and indefinitely keeping the node's radio awake.

To execute an efficient replay attack, an attacker must be able to closely estimate the value of *sleepTime* in replayed SYNC packets. If traffic is authenticated, but not encrypted, the attacker can read (but not modify) the *sleepTime* value and send SYNC packets at a rate slightly less than $\lfloor sleepTime/2\rfloor$. If packets are encrypted and the attacker cannot read the value of *sleepTime* in the recorded packet, the attacker can mount an efficient attack by assuming a minimum value for *sleepTime* in a SYNC packet. The value of *sleepTime* is set based on the value of a decrementing counter that begins when the node transitions from the sleep portion of a frame to the SYNC period in the next frame. On the Mica2 implementation using default S-MAC settings, this counter is initialized to 130 and decrements every 1 ms. Each node checks whether it should send a SYNC packet at the beginning of the SYNC period and, if so, begins to backoff for the duration of a DIFS (10 ms) plus a random number of milliseconds within its 15-ms contention window, for a maximum backoff of 25 ms. Using these values, the minimum value of *sleepTime* is 105 ms. Experiments were conducted under the conditions shown in Fig. 6 using a rate of 50 ms between packets, which is slightly less than half of the minimum assumed *sleepTime* of 105 ms. This SYNC interval proved to be effective, and all nodes except the cluster head were constantly kept awake. The cluster head node is not kept awake because it disregards the counterfeit SYNC packet with its own address as the source.

The S-MAC implementation for Mica2 uses an effective data rate of 19.2 kb/s using Manchester encoding [3]. At that data rate, a SYNC packet, including preamble and all headers, takes 13.6 ms to send. Mica2 requires 2.6 ms of transition time from transmit mode to sleep mode and back to transmit mode [6]. The attacker's radio must, therefore, be awake for 16.2 ms of every 50.0 ms, or 32% of the time.

3) Class 3—Full Domination Attack: In a full domination attack, an attacker is assumed to be able to set the value of *sleepTime* in an S-MAC SYNC packet, either because the network is not using encryption or authentication or because the attacker has broken the encryption mechanism. The *sleepTime* in a SYNC packet is 2 B; therefore, the maximum value that could be used as the *sleepTime* is 65 535 ms. An attacker cannot, however, mount an effective attack by simply setting the *sleepTime* to this maximum value and sending a SYNC packet at a rate of $\lfloor 65\,535/2\rfloor$ ms. This is because the bogus SYNC packets are sent using the same node identification number as the cluster head so that the victim nodes do not interpret the SYNC as a new additional schedule in the network but as a modification to their primary schedule. By setting a *sleepTime* that is longer than the frame length, none of the nodes in the cluster except the cluster head is able to enter sleep mode. To maintain this attack, the attacking node must send out its forged SYNC packet after each SYNC sent by the cluster head. The default S-MAC configuration has the cluster head sending a SYNC packet every tenth frame, or every 13 000 ms. The full domination attack was implemented using these parameters and using a *sleepTime* value of 6000 ms, which is longer than the frame duration. As expected, all the attacked nodes were kept awake 100% of the time, except for the cluster head node, which maintained its default 10% duty cycle.

Since the attacker is awake during every tenth duty cycle to receive the cluster head's SYNC packet and send its own, the attacking node is awake for 130 ms out of every 13 000 ms, or 1% of the time. Note that the SYNC attacks described here rapidly drain the batteries of all nodes except for the cluster head nodes. Once all the noncluster head nodes have been exhausted, the remaining nodes will form clusters if they are within communication range and can be attacked as described above. If the remaining nodes cannot form clusters because they are too far apart to communicate, the network is no longer usable.

B. Attacks on T-MAC

1) Class 1—Constant Jamming Attack: A constant jamming attack using a signal generator has the same effect on T-MAC as on S-MAC. Nodes are prevented from exchanging traffic, but they are not prevented from placing the radio in sleep mode after the adaptive timeout has expired. The deceptive jamming attack, however, forces the victim nodes to reset their time to sleep based on the TA value after each received start symbol. The deceptive jamming attack, therefore, keeps the victim devices awake 100% of the time. While effective in terms of a denial-of-sleep attack, it is not efficient in that the attacker must be awake and transmitting 100% of the time and could potentially deplete its energy supply before the victim nodes.

2) Class 2—Adaptive Timeout Attack: A simpler attack on T-MAC that takes advantage of knowledge of the protocol's functionality exploits the adaptive timeout mechanism described in Section II-B. By sending a small packet at an interval just short of the network's adaptive timeout, sensor devices on a T-MAC network will remain constantly awake. The TA interval is platform dependent and, as shown in (2), is based on the maximum contention window duration, the duration of an RTS message, and the duration of a SIFS. On the Mica2 platform implementation, which was written by the original designers of T-MAC, the authors use a slightly different formula for calculating TA than the one described in [5] [which is repeated in (2)]. Instead of using T (the SIFS duration), they use 2× the time that a node should wait for a CTS message after receiving an RTS message, which is calculated as the amount of time required to send a 12-B CTS packet, plus a 10-ms processing delay. The value of TA, therefore, becomes

$$TA = 1.5 \times (C + (2 \times T_{Wait\_CTS})) . \tag{7}$$

For the Mica2 implementation, the adaptive timeout duration is $TA = 81.6$ ms.

To confirm the effectiveness of an attack taking advantage of the adaptive timeout mechanism, a network running the T-MAC protocol was attacked by sending a stream of 1-B-long packets. The attack used an interval of 70 ms between packets, which is 10 ms shorter than the TA value to account for processing, transmission, and propagation delays. Unfortunately, despite significant debugging, the T-MAC testbed on which this attack was implemented maintained a higher-than-normal packet loss rate of 20%–25% both for routine T-MAC traffic and for attack traffic. The attack at 70 ms was, therefore, not as effective as expected, keeping the network awake an average of 83% of the time. By setting the attack interval to 35 ms, the attack effectiveness increased, keeping targeted nodes awake an average of 92% of the time. To determine the effectiveness of the adaptive timeout attack on a network with perfect physical-layer properties, the 70-ms attack was carried out on a simulated network using the *Avrora* [21] emulation environment.[2] Avrora emulates the Mica2 platform on a PC and allows the user to directly run the TinyOS code that is compiled for Mica2. On the Avrora platform, the adaptive timeout attack worked as expected, keeping the victim devices awake 100% of the time.

It takes the attacker 2.9 ms to send a 1-B packet, along with preamble. With a total of 2.6 ms to wake up before sending the packet and to go to sleep afterward, the attacking node must be awake 5.5 ms every 70 ms, or 8% of the time.

3) Class 3—Full Domination Attack: While T-MAC uses SYNC messages that specify *sleepTime* much like S-MAC does, this mechanism cannot be exploited to keep nodes awake in the same way it can be done in an S-MAC network. The T-MAC SYNC messages help to keep nodes synchronized and inform nodes of other active schedules, but the time that a node is awake is still dictated by the adaptive timeout (TA). Therefore, the most efficient denial-of-sleep attack on T-MAC is the adaptive timeout attack described in the previous section. It should be noted that to mount this attack against T-MAC, it is simply necessary to know that the network is running the T-MAC protocol. No penetration of link-layer encryption or authentication is required.

C. Attacks on B-MAC

1) Class 1—Constant Jamming Attack: A constant jamming attack using the signal generator has the same affect on B-MAC as on the other protocols. B-MAC regularly samples the wireless channel and uses the previous n samples of idle channel time to determine a threshold value for background noise. Constant jamming is, therefore, recognized as background noise and is ignored. As a result, this jamming prevents nodes from exchanging packets, but it does not prevent them from entering sleep mode. Under the experimental configuration, nodes are awake to conduct an LPL sample for approximately 2.6 ms out of every 135 ms, for an awake percentage of 2%. Unlike constant jamming, deceptive jamming prevents B-MAC nodes from entering sleep mode. As soon as a B-MAC node awakens and recognizes a preamble being transmitted, it begins a cycle of receiving bytes and searching for a pair of *synchronization bytes* that indicate the beginning of the expected data packet. The victim node will indefinitely sample incoming preamble bytes until a packet is observed, remaining awake 100% of the time. This attack is very effective as a denial-of-sleep attack, but the attacker must also be active 100% of the time, making it inefficient. It is particularly costly for the attacker on the Mica2 platform since the power consumption during data transmission is approximately twice that of receive-mode power consumption. On the Tmote Sky platform, the transmit cost is lower than the receive cost, making this a more practical attack.

2) Class 2—Unauthenticated Broadcast Attack: The unauthenticated broadcast attack described in Section V-B is achieved by sending a constant stream of back-to-back broadcast packets. An attacker might also record a maximum-size broadcast packet and replay this packet into the network. Both attacks have the same denial-of-sleep impact. Table IV gives the expected B-MAC network lifetime under such an attack, as determined using (4). In practice, however, the percentage of victim node awake time was higher than that predicted by (4). This is because in the Mica2 B-MAC implementation, nodes do not immediately transition to sleep mode when packet reception is complete. Instead, they remain in receive mode, only transitioning to sleep mode when a periodic timer fires and a check is made to determine whether the node is in transmit, receive, or idle mode (not actively sending or receiving packets). Under this attack, victim nodes were kept awake an average of 79% of the time, making it much more effective than expected. Since the attacker must transmit 100% of the time, however, it is not as efficient an attack as constant deceptive jamming. The advantages of this attack over deceptive jamming are twofold. First, the attacker can follow MAC-layer collision avoidance rules and only transmit when the channel is idle, thus allowing legitimate traffic to be transmitted and increasing the subtlety of the attack. Second, this attack would not be recognized as a jamming attack if the network were protected from jamming using techniques described by Xu *et al.* [16].

3) Class 3—Full Domination Attack: The most effective attack against B-MAC is the deceptive jamming attack. Despite its inefficiency, this attack keeps victim nodes awake 100% of the time. It is also easy to execute. Simply recognizing that B-MAC is being used in a network is all the information an attacker needs to keep network nodes in constant receive mode, rapidly draining their energy supplies.

D. Attack Efficiency

Table V shows the efficiency of the various denial-of-sleep attacks on S-MAC, T-MAC, and B-MAC described in this section. The *Active Ratio* (*AR*) is the ratio of the percentage of time that the attacker is awake to the percentage of time that the victim nodes are awake during the period that the attack is active. This metric is used because of the disparity between transmit and receive power consumption among sensor platforms. For example, for Mica2, the transmit cost is twice that of the receive cost, whereas for Tmote Sky, the receive cost is slightly higher than the transmit cost. Assuming equivalent energy consumption, *AR* is the unit of energy expended by the attacker to deplete one unit of energy from the victim(s). An *AR* value of less than 1.0 means that the victim nodes are awake for a higher percentage of time than the attacker. Constant physical-layer jamming and deceptive jamming are

[2] Tests with Avrora were performed using version 1.7.59, which was the version available in the Avrora CVS repository [19] as of August 1, 2006.

TABLE V
EFFICIENCY OF DENIAL-OF-SLEEP ATTACKS

		S-MAC		
Attack Class	Attack	Attacker Awake	Victim Awake	Active Ratio (AR)
I	PHY jamming	100%	10%	10.0
I	Deceptive jamming	100%	10%	10.0
II	SYNC replay	32%	100%	0.32
III	Efficient SYNC	1%	100%	0.01
		T-MAC		
Attack Class	Attack	Attacker Awake	Victim Awake	Active Ratio (AR)
I	PHY jamming	100%	6%	17.0
I	Deceptive jamming	100%	100%	1.0
II	TA attack (Mica2)	8%	83%	0.09
II	TA attack (Avrora)	8%	100%	0.08
III	TA attack (Avrora)	8%	100%	0.08
		B-MAC		
Attack Class	Attack	Attacker Awake	Victim Awake	Active Ratio (AR)
I	PHY jamming	100%	2%	50.0
I	Deceptive jamming	100%	100%	1.0
II	Broadcast attack	100%	79%	1.27
III	Deceptive jamming	100%	100%	1.0

particularly inefficient when mounted against an S-MAC network. Although deceptive jamming attack permanently keeps T-MAC nodes awake, it requires the attacker to be constantly awake and is, therefore, not an efficient attack on T-MAC either. There is no energy-efficient denial-of-sleep attack against B-MAC. The most efficient attack against B-MAC is the constant deceptive jamming attack, which keeps both victims and attackers awake 100% of the time. The efficiency of attacks such as the full domination attack on S-MAC makes clear the necessity for strong link-layer encryption in deployed sensor networks. The attacks presented in which the nature and parameters of the MAC protocol are known but link-layer encryption is *not* compromised are less efficient; however, they show that an attacker can still use a mote-class device to quickly drain the energy reserves of sensor devices using S-MAC or T-MAC with significant power to spare.

E. Discussion

The relative ease with which the denial-of-sleep attacks described in this section were implemented and the efficiency with which they can be carried out indicate that energy-efficient MAC protocols must be designed with security in mind. In the cases described above, the very mechanisms used in these protocols to conserve energy are exploited to rapidly drain power on devices that use them.

VII. FRAMEWORK FOR DEFENDING AGAINST DENIAL-OF-SLEEP ATTACKS IN WSN

In this section, a framework for defending against denial-of-sleep attacks is presented. To prevent attacks across the spectrum of link-layer vulnerabilities, a defensive framework must incorporate four key components, i.e., strong link-layer authentication, anti-replay protection, jamming identification and mitigation, and broadcast attack defense.

1) Strong Link-Layer Authentication: This is the first and most important component of denial-of-sleep defense and must be incorporated into any WSN that might be vulnerable to attack. Authentication at higher protocol layers can be effective for providing data integrity and confidentiality but still fails to ensure service availability. An attacker's ability to send trusted MAC-layer traffic on the network leaves it open to the types of full-domination attacks that can reduce the network lifetime from a year or more to less than a week. Existing options for implementing link-layer authentication in WSN include TinySec, which is incorporated into current releases of TinyOS [20], and the authentication algorithms built into IEEE 802.15.4-compliant devices.

2) Anti-Replay Protection: An attacker's ability to replay messages, even without being able to read them, can force nodes to forward old traffic through the network and can significantly increase power consumption for all nodes on the path from sender to receiver. Traffic analysis makes it possible to distinguish control traffic from data traffic. Replayed control packets, like S-MAC SYNC packets, can be used to mount an effective denial-of-sleep attack.

Existing techniques for protecting against replay attacks at the link layer have the disadvantage of requiring resource-constrained sensor nodes to maintain a neighbor table of packet sequence numbers, a requirement that can become unwieldy even in moderately sized networks. The neighbor table can also be exploited by an attacker if packets from other portions of the network are replayed, thereby increasing the size of a node's neighbor table and consuming more resources. One way to limit the size of the neighbor table is to use network-layer neighbor information to limit the number of neighbors that must be tracked to those from which legitimate traffic is expected. Clustering protocols such as HEED [22] and ACE [23] reduce the number of potential communication partners to a subset of a node's one-hop neighbors. By adding a small amount of anti-replay information to clustering messages and using existing authentication techniques, anti-replay protection can be provided for clustered WSNs at low overheads. One such technique is Clustered Anti-replay Protection (CARP), as described in [24]. CARP bounds the size of the neighbor table according to the maximum node degree and the number of clusters, which are user configurable in many clustering protocols. Anti-replay counters are exchanged during the periodic reclustering process. This anti-replay counter exchange is, in turn, protected from replays using a sequential numbering scheme for clustering events. Since reclustering is, by definition, a network-wide operation, all nodes know the sequence number of the current clustering event, and replayed clustering messages from previous clustering events can be identified and ignored [24].

3) Jamming Identification and Mitigation: A strong jamming attack can prevent all sensor nodes' access to the wireless medium and can shut down the network. To reduce costs, sensor nodes are usually equipped with simple radios that are not designed to use spread-spectrum techniques to defend against jamming. While IEEE 802.15.4-compliant transceivers use direct sequence spread spectrum (DSSS) to protect against background noise, spreading codes are fixed according to the

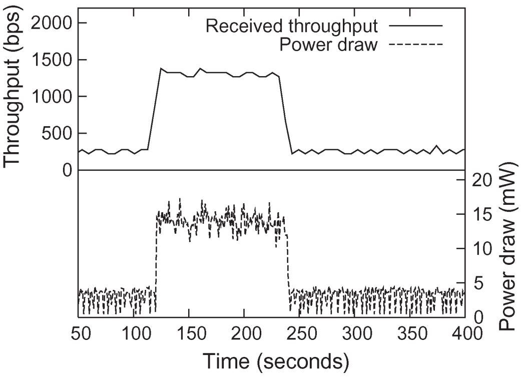

Fig. 7. Received throughput and receiver per-second power consumption for a WSN node during a burst of network traffic.

ZigBee standard and, therefore, cannot be used to defend against jamming by a ZigBee-compliant attacker. A logical reaction to jamming is for nodes to go into low-power mode, waking only periodically to sense the medium, thus conserving maximum energy when there is no hope of successfully using the wireless medium. With techniques available to reliably identify jamming attacks, such a mechanism is now feasible. As part of this research, Xu *et al.*'s proposed jam detection mechanism based on the relationship between PDR and RSSI values [16] was implemented and tested on the Mica2 WSN platform. This implementation effectively detects jamming with a low probability of false positives. Adding jam detection to networks that are vulnerable to jamming-based denial-of-sleep attacks is quite possible using this technique.

4) Broadcast Attack Protection: Most MAC protocols are susceptible to a simple unauthenticated broadcast attack. Long messages can be broadcasted and must be received in full by all network nodes before the nodes discard them due to authentication failure. A *subtle broadcast attack* is one in which the attacker obeys MAC-layer rules of collision avoidance, thereby transmitting attack traffic only when there is no legitimate traffic in the network. This type of attack is particularly hard to detect because it does not effect legitimate throughput, which might indicate an ongoing network attack. The limited resources available on most sensor platforms prevent the use of traditional network intrusion detection techniques, which normally require capturing and analyzing large amounts of previous network traffic. Another alternative, however, is a lightweight intrusion-detection mechanism employed at the MAC layer that classifies each incoming packet as either *legitimate* (meaning that it passes authentication and anti-replay checks) or *malicious*. Tracking the ratio of legitimate to malicious traffic, along with the percentage of time that the device is able to sleep, is enough to identify a denial-of-sleep broadcast attack [25]. Fig. 7 shows the correlation between received traffic and power consumption in a simulated Mica2 network. The offered load averages 1 packet-per-second (pps) with a burst of 4 pps of legitimate traffic from 120 to 240 s. Since this burst is legitimate data, it should be allowed despite increased power consumption during the burst. As long as legitimate traffic can be differentiated from malicious traffic, the spike in energy consumption associated with the increase in traffic, along with a high ratio of malicious versus legitimate traffic, identifies the requirement to take action to mitigate the energy-draining effects of malicious traffic.

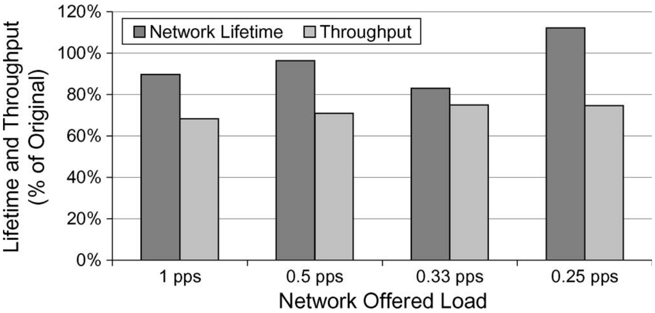

Fig. 8. Percent of original network lifetime and throughput maintained using the automated rate-limiting mechanism in B-MAC across network-offered loads.

Once an attack is identified, rate limiting can be triggered to reduce energy consumption. Normally, rate limiting places limits on outgoing traffic to reduce network congestion or to provide quality-of-service guarantees. Since a malicious node cannot be forced to limit the amount of traffic it transmits, rate limiting, in this case, limits the amount of time that nodes are awake to receive traffic. For example, in a T-MAC network, the number of times that a node's sleep timer is reset during a frame might be reduced based on high levels of malicious traffic and low sleep ratios.

A description of how this technique can be applied to networks using the B-MAC protocol is found in [25], where the adaptive rate-limiting mechanism is simulated in OPNET [26]. Fig. 8 depicts the percentage of network lifetime and broadcast throughput maintained across network-offered loads using the automated rate-limiting mechanism when a subtle broadcast attack is mounted against the network. Even during a long-term attack, over 80% of the original expected network lifetime is preserved while maintaining 77% or better legitimate network throughput, depending on the network-offered load. If the broadcast attacker is not one that obeys MAC protocol collision avoidance rules, the network lifetime can still be maintained, although throughput cannot be for the duration of the attack.

VIII. Conclusion and Future Work

Most current research in WSN security focuses on data confidentiality and integrity, largely ignoring availability. Without the ability to secure the physical medium over which communication takes place, sensor networks are susceptible to an array of potential attacks focused on rapidly draining sensor node batteries, thereby rendering the network unusable. This paper makes three contributions to the area of sensor network security. First, it classifies denial-of-sleep attacks on WSN MAC protocols based on an attacker's knowledge of the MAC protocol and ability to penetrate the network. Second, it explores potential attacks from each attack classification, both modeling their impacts on sensor networks running four leading WSN MAC protocols and analyzing the efficiency of implementations of these attacks on three of the protocols. Finally, it proposes a framework for defending against denial-of-sleep attacks and provides specific techniques that can be used against each denial-of-sleep vulnerability. Future work will involve exploring the defensive framework provided here and finding ways to apply it to currently available sensor devices to further develop specific mechanisms to protect them against these attacks.

Acknowledgment

The authors would like to thank the anonymous reviewers, whose comments helped to improve this paper.

References

[1] M. Brownfield, Y. Gupta, and N. Davis, "Wireless sensor network denial of sleep attack," in *Proc. 6th Annu. IEEE SMC Inf. Assurance Workshop*, Jun. 2005, pp. 356–364.

[2] *Tmote Sky Datasheet: Low Power Wireless Sensor Module,* Moteiv Corporation, Redwood City, CA. Accessed Feb., 2006. [Online]. Available: http://www.moteiv.com/

[3] *Mica2 Datasheet,* CrossBow Corporation, San Jose, CA. Accessed May 2006. [Online]. Available: http://www.xbow.com/

[4] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[5] T. VanDam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. 1st ACM Int. Conf. Embedded Netw. Sensor Syst.*, Nov. 2003, pp. 171–180.

[6] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. 2nd ACM Int. Conf. Embedded Netw. Sensor Syst.*, Nov. 2004, pp. 95–107.

[7] M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis, "Wireless sensor network energy-adaptive MAC protocol," in *Proc. IEEE Consum. Commun. Netw. Conf.*, Jan. 2006, pp. 778–782.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, pp. 8020–8029.

[9] G. Pei and C. Chien, "Low power TDMA in large wireless sensor networks," in *Proc. AFCEA/IEEE Military Commun. Conf.*, Oct. 2001, pp. 347–351.

[10] S. Singh and C. S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," *Comput. Commun. Rev.*, vol. 28, no. 3, pp. 5–26, Jul. 1999.

[11] M. Brownfield, N. Davis, and A. Fayez, "Wireless sensor network radio power management," in *Proc. OPNETWORK*, Aug. 2005.

[12] A. Perrig, R. Canetti, D. Song, and J. Tygar, "Efficient and secure source authentication for multicast," in *Proc. 8th Annu. Symp. Netw. Distrib. Syst. Security*, Feb. 2001, pp. 35–46.

[13] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.*, Nov. 2004, pp. 162–175.

[14] LAN MAN Standards Committee of the IEEE Computer Society, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4, 2003.

[15] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[16] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw.*, May 2005, pp. 46–57.

[17] R. Negi and A. Perrig, "Jamming analysis of MAC protocols," Carnegie Mellon Univ., Pittsburgh, PA, 2003. Tech. Rep.

[18] Y. W. Law, L. vanHoesel, J. Doumen, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," in *Proc. 3rd ACM Workshop Security Ad Hoc Sensor Netw.*, Nov. 2005, pp. 76–88.

[19] SourceForge.net. Accessed Jun. 2006. [Online]. Available: http://sourceforge.net/

[20] *TinyOS Community Forum,* Accessed Aug., 2007. [Online]. Available: http://www.tinyos.net/

[21] *Avrora—The AVR simulation and analysis framework.* Accessed Aug., 2006. [Online]. Available: http://compilers.cs.ucla.edu/avrora/

[22] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Dec. 2004.

[23] H. Chan and A. Perrig, "ACE: An emergent algorithm for highly uniform cluster formation," in *Proc. 1st Eur. Workshop Sensor Netw.*, Jan. 2004, pp. 154–171.

[24] D. Raymond, R. Marchany, and S. Midkiff, "Scalable, cluster-based anti-replay protection for wireless sensor networks," in *Proc. 8th Annu. IEEE SMC Inf. Assurance Workshop*, Jun. 2007, pp. 127–134.

[25] D. Raymond and S. Midkiff, "Clustered adaptive rate limiting: Defeating denial-of-sleep attacks in wireless sensor networks," in *Proc. AFCEA/IEEE Military Commun. Conf.*, Oct. 2007, pp. 1–7.

[26] *OPNET Modeler,* Bethesda, MD: OPNET Technol. Inc. Accessed Aug. 2006. [Online]. Available: http://www.opnet.com/

David R. Raymond (S'05–M'08) received the B.S. degree in computer science from the United States Military Academy, West Point, NY, the M.S. degree in computer science from Duke University, Durham, NC, and the Ph.D. degree in computer engineering from Virginia Polytechnic Institute and State University, Blacksburg.

He is currently a Lieutenant Colonel with the United States Army stationed at Fort Leavenworth, KS. His research interests include energy-efficient MAC protocols for wireless sensor networks, mobile and ad hoc networking, and network security.

Mr. Raymond is a Student Member of the Association for Computing Machinery (ACM).

Randy C. Marchany (A'93) received the M.S. degree in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg.

He is the Director of the IT Security Lab, Virginia Polytechnic Institute and State University, Blacksburg, which is a component of the university's Information Technology Security Office. He is the Coordinator of VA-CIRT, which is an incident response team comprised of IRTs from various Virginia State Universities. He has been a member of the SANS Institute's faculty since 1992 and is one of the developers of their GIAC security certification courses.

Mr. Marchany was a recipient of the SANS Institute's Security Technology Leadership Award for 2000, the Virginia Governor's Technology Silver Award in 2003, and an EDUCAUSE award in 2005.

Michael I. Brownfield (M'06–SM'08) received the B.S. degree in electrical engineering from the United States Military Academy, West Point, NY, in 1989, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1999, and the Ph.D. degree in electrical engineering from Virginia Polytechnic and State University, Blacksburg, in 2006.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, United States Military Academy, and a Lieutenant Colonel with the United States Army. His research interests include wireless sensor networks and mobile ad hoc networks.

Dr. Brownfield is a Licensed Professional Engineer.

Scott F. Midkiff (S'82–M'85–SM'94) received the Ph.D. degree in electrical engineering from Duke University, Durham, NC.

He has been on assignment as Program Director with the U.S. National Science Foundation since September 2006. He is currently a Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg. His research interests include system issues in wireless, sensor, and ad hoc networks, network services for pervasive computing, and performance modeling of mobile ad hoc networks.