Instrumenting Network Simulators for Evaluating Energy Consumption in Power-Aware Ad-Hoc Network Protocols

Cintia B. Margi*, Katia Obraczka
University of California Santa Cruz
1156 High Street
Santa Cruz, CA 95060

Abstract

In this paper, we describe our work on instrumenting network simulators to enable them to adequately and accurately account for the energy consumed by ad hoc network protocols' communication-related tasks. This is accomplished by explicitly accounting for low-power radio modes and considering the different energy costs associated with each possible radio state, i.e, transmitting, receiving, overhearing, idle, sensing, and sleeping. Our energy consumption instrumentation also allows the energy accounting to be done automatically by the simulator irrespective of what layer of the stack the protocol designer is working. To validate our model, we compare (1) simulation results using the GloMoSim/QualNet simulation platform with and without our instrumentation for the IEEE 802.11 DCF, (2) analytical results for both 802.11 and S-MAC (a power-aware MAC designed for sensor networks), and (3) simulation results reproducing testbed experiments obtained for the S-MAC protocol. Finally, by comparing S-MAC against 802.11 and AODV against DSR, we showcase the ability of a network simulation platform instrumented with our energy consumption model to evaluate energy consumption in ad-hoc network protocols.

1. Introduction

Wireless ad-hoc networks are also known as "networks without a network" since they do not use any fixed infrastructure. Participating nodes in these networks are typically battery operated, and thus have access to a limited amount of energy. Frequently, once nodes are deployed, their batteries cannot be easily recharged. Sensor network nodes are a typical example as some of them have very limited battery life; moreover, once deployed, a sensor network may be left unattended for its entire operational lifetime. This is due to the fact that sensor networks may be deployed in wide, remote, unaccessible areas.

The energy-constrained nature of ad hoc networks in general, and sensor networks in particular, calls for protocols that have energy efficiency as a primary design goal. Research on power-aware protocols has been very active and spans multiple layers of the protocol stack. As a result, several energy-efficient medium-access control (MAC)- and routing protocols have been proposed.

In order to evaluate and compare power-aware protocols in terms of their energy efficiency as well as assess the effectiveness of cross-layer mechanisms to achieve energy savings, accurately accounting of the energy consumed by data communication activities is crucial. Such accounting must be as close to reality as possible, taking into consideration all radio states, i.e., energy spent not only while transmitting and receiving a packet, but also while in idle, overhearing, or sleep modes. Frequently, the evaluation of network protocols is carried out using network simulators such as `ns-2` [11], `GloMoSim` [27], and `QualNet` [19]. As explained in detail in Section 2, the models used by these simulators to account for energy consumption by data communication activities are not accurate. More specifically, the models employed either do not consider all radio states or do not take into account the different energy levels they consume. Furthermore, most current simulators do not automatically measure energy consumption, leaving it up to the protocol designer to explicitly write code to account for it. And, clearly, depending on the layer of the protocol stack, energy consumption accounting can become quite cumbersome and inaccurate. Not to mention the duplication of effort as code to accomplish the same task is written several times for the same simulation platform.

The contribution of this paper is two-fold. First, it describes our work on instrumenting network simulators to enable them to adequately and accurately

* This author is supported by CNPq/Brazil.

account for the energy consumed by ad hoc network protocols' communication-related tasks. This is accomplished by explicitly accounting for low-power radio modes and considering the different energy costs associated with each possible radio state, i.e, transmitting, receiving, overhearing, idle, sensing, and sleeping. Second, our energy consumption instrumentation also allows the energy accounting to be done automatically by the simulator irrespective of what layer of the stack the protocol designer is working. For example, in [3], the analytical model presented for energy consumption in IEEE 802.11 single-hop wireless networks is compared to the accounting provided by QualNet with our energy consumption instrumentation. The results match quite closely differing by at most 15%. We then validate our energy consumption model by comparing (1) simulation results using the GloMoSim/QualNet [19] simulation platform with and without our instrumentation for the IEEE 802.11 DCF [4], (2) analytical results for both 802.11 and S-MAC [26] (a power-aware MAC designed for sensor networks), and (3) simulation results reproducing testbed experiments obtained for the S-MAC protocol [26]. Finally, we showcase the ability of a network simulation platform instrumented with our energy consumption model to evaluate power-aware protocols by comparing S-MAC against 802.11. We also evaluate the energy consumption of AODV [15] and DSR [12]. Although we implement our energy consumption instrumentation on the GloMoSim/QualNet platform, it can easily be ported to any existing network simulators (e.g., ns-2 [11]).

The remainder of this paper is organized as follows. Section 2 reviews related work. In Section 3, we describe our energy consumption instrumentation. Our experimental methodology and model validation results are presented in Section 4. Results comparing MAC protocols, namely S-MAC and 802.11, and routing protocols, namely AODV and DSR, are presented in Section 5 and our concluding remarks and directions for future work in Section 6.

2. Related Work

2.1. Energy models and energy consumption in Network Interfaces

In [21] the power consumption of some network interface cards (NICs) was measured when used by different end-user devices. They also report on transport- and application-level strategies to reduce energy consumption by NICs. Later, [7] reported detailed energy consumption measurements of some commercially-available IEEE 802.11 NICs operating in ad hoc mode. Energy consumption models using linear equations were also introduced. For example, $E = m * size + b$ models data transmission and reception, where the coefficients m and b depend on the type of communication, i.e., broadcast, unicast, or packet discarded, and were determined empirically.

Along the same lines, [6] assessed the impact of transmission rate, transmit power, and packet size on energy consumption in a typical wireless network interface. In most previous measurements, however, the focus was on the characterization of energy consumption during the many modes of operation of a NIC (transmit, receive, idle, etc.), under extremely simple scenarios, e.g., only two nodes operating in ad hoc mode, with one node acting as the sender and the other as the receiver.

In [1], a power consumption model for sensor networks consisting of three components (1)sensor-, (2)computation-, and (3)communication cores is proposed. The sensor core, which assumes that the energy used to sense a bit is constant, defines power consumption as $P_{sense} = \alpha_3 * r$, where r is the bit-rate and a typical value for α_3 is 50 nJ/bit. Power consumed in the computation core is attributed to data aggregation and is defined by $P_{comp} = \eta_{agg} * \alpha_4 * r$, where α_4 can vary from a few pJ/bit to 10 nJ/bit, and η_{agg} is the number of streams being aggregated. Finally, the communication model has two portions: transmission and reception. Power consumption due to reception is proportional to r, the bit rate, as shown in $P_{rx} = \alpha_{12} * r$, where α_{12} is 135 nJ/bit. Power spent on transmission, given by $P_{tx}(n_1, n_2) = (\alpha_{11} + \alpha_2 * d(n_1, n_2)^n * r)$, also depends on the bit rate r, as well as the distance between nodes $(d(n_1, n_2))$ and path loss index n. However, no validation of the energy model is presented in this work. Moreover, energy spent in data processing is attributed to data aggregation and does not account for local processing (e.g., data compression).

An energy-aware simulation model, which considers a network consisting of multiple nodes, where each node is composed by a local request queue, a microprocessor, an external request queue, another processor, a service queue and a service provider is describe in [22]. All components are random variables. The total energy consumption on a node is the sum of the energy spent by node components, energy consumption for transmitting a data packet, and energy consumption for receiving a data packet. Although the model considers energy spent with processing and buffering requests, the radio model is quite simple and does not include a low-power radio mode, which is crucial for development of power-aware protocols.

A simple energy model is introduced in [9] to evaluate power-aware protocols in the LEACH project [8]. LEACH (Low-energy Adaptive Clustering Hierarchy) is a clustering-based protocol that uses randomized rotation of cluster-heads to evenly distribute the energy load among the sensors in the network. In the energy model, the energy spent on transmission is given by the energy dissipated by the radio electronics and the power amplifier, while the energy spent by the receiver is given by the the energy dissipated by the radio electronics. Since the energy necessary to amplify the signal depends on its attenuation, and the attenuation depends on the distance, the energy dissipated by the radio electronics is proportional to d^2 for short distances and to d^4 otherwise. Using this same energy model, [5] examine the energy consumption in a wireless sensor network with two distinguished organizations: single layer versus clustered.

In [14], energy consumption in ad-hoc mobile terminals is modeled using the Advanced Configuration Power Interface [10], or ACPI, an open standard that allows computer systems to implement motherboard configuration and power management functions. ACPI was used to measure energy consumption due to transmission/reception. The resulting energy consumption model includes two states: **high consumption state**, where the host receives and transmits, and **low consumption state**, where the node receives or is in idle. While this approach to model battery discharge empirically is based on values that laptop power management would see in real systems, it is platform-dependent.

A comparison of transmitter's energy consumption of TCP Reno, Newreno and SACK using a laptop testbed under different network conditions (e.g.,variable round trip times, random loss, bursty loss, packet reordering, etc.) is presented in [20]. The results shows that SACK consumes the least total energy, but if the energy cost for idle states is much lower than receiving state, it performs poorly.

A sensor networks testbed of PicoNodes [17] is used to evaluate energy consumption on different radio states and different traffic types.

Developers of power-aware protocols often implement their own energy models (e.g., [16], [13]) at the layer of the protocol stack they are working on. An alternative is to use what is available in current network simulators or add extensions to obtain the desired results. Thus results from different efforts cannot be compared directly. For example, [25] does an evaluation of topology control protocols using `ns-2` and the parameters for the radio are from [21].

An analytical model to predict energy consumption in saturated IEEE 802.11 single-hop ad hoc networks under ideal channel conditions is presented in [3]. The energy consumption predicted by the model is then compared to the accounting provided by instrumented QualNet. Important results from this work include the following: (1) contrary to what most previous results indicate, the radio's *transmit* mode has marginal impact on overall energy consumption, while other modes (*receive, idle, etc.*) are responsible for most of the energy consumed; (2) the energy cost to transmit useful data increases almost linearly with the network size; and (3) transmitting large payloads is more energy efficient under saturation conditions.

2.2. Network simulators

GloMoSim and QualNet In QualNet 3.6 [19], the energy consumption model for communication is implemented in the physical layer. The simulator currently includes six different physical layer models: 802.11a, 802.11b, abstract, GSM, FCSC prototype, and link16.

The current radio model only defines four states: idle, sensing, receiving (RX) and transmitting (TX); there is no state corresponding to the low-power energy mode where the radio cannot transmit or receive (usually referred as "sleep" state). For energy consumption purposes, QualNet considers that the radio is either in TX or RX states (in ad hoc network mode). If the radio is in RX, it spends 900 mW. The power consumption for transmitting signals is calculated as: $(TxPowerCoef * txPower + TxPowerOffset) * txDuration$

The values of $TxPowerCoeff$ and $TxPowerOffset$ are statically defined based on the WaveLAN specifications, and are assigned the values of 16/sec, and 900mW (the same value as consumed in RX mode). $txPower$ is proportional to the distance the signal is supposed to travel. For each frame transmitted, the energy spent is calculated and added to the energy consumption statistics variable. Once the simulation ends, total simulation time is multiplied by the cost of being in RX mode and added to the energy consumption statistics.

Clearly, QualNet's current energy model is not realistic enough as it does not distinguish between RX, overhearing and idle states. Furthermore, it does not include a low-power, sleep state. Another drawback of QualNet is that there is no energy consumption information during simulation time. The amount of energy consumed is only available at the end of a simulation run. For the purpose of evaluating energy consumption, this is not a real issue, but if the goal is to simulate nodes failing when running out of battery and/or have energy-aware protocols looking at energy information to make certain decisions, this model is not adequate.

GloMoSim [27], QualNet's precursor provides an energy model that is very similar to QualNet's.

ns-2 The energy model supported by `ns-2` [11] includes four states: idle, sleep, receiving (RX) and transmitting (TX). Every node starts with an initial energy level and consumes energy as it transmits and receives data. Periodically, nodes update the amount of energy spent in idle state.

Energy consumption for TX, RX, and idle states have default values of $Pt_{consume} = 0.660$, $Pr_{consume} = 0.395$ and $Pi_{consume} = 0.0$, respectively. However, to our knowledge, there is no mention of the energy consumption in sleep state. While in sleep state, `ns-2` keeps an accounting of the time spent in this state. There is a note in the documentation that says that "time in the sleep mode should be used as credit to idle time energy consumption", which implies that $Ps_{consume} = 0.0$.

Some of the weaknesses of the current energy model employed by `ns-2` include the lack of calculation of the energy consumed in sleep state which does not allow a fair comparison between protocols that explicitly use this mode for power savings. A minor drawback is that, if the user does not set the value of $Pi_{consume}$, the total energy consumption will reflect only what was spent in RX and TX.

2.3. Power-aware MAC protocols

In this paper, we use two MAC protocols, namely IEEE 802.11 DCF [4] and S-MAC [26], to validate and showcase the proposed instrumentation. We describe their main features below.

The IEEE 802.11 DCF [4] is a contention-based protocol based on carrier sensing with collision avoidance (CSMA/CA). IEEE 802.11 performs both physical and virtual carrier sensing. Virtual carrier sensing is achieved by sending information about duration of each frame in the headers which is used by stations as an indication of how long the channel will be busy. After this time is elapsed, stations can sense the channel again. In order to solve the "hidden terminal" problem and avoid data frame collisions, the RTS-CTS handshake is used. Two power management mechanisms are supported: *active* and *power-saving* (PS).

The Sensor MAC protocol [26], or S-MAC, was developed with power savings as one of its design goals. It also falls into the contention-based protocol category but achieves energy efficiency by making use of low-power radio mode. Nodes alternate between periodic sleep and listen periods. Listen periods are split into synchronization and data periods. During synchronization periods, nodes broadcast their sleeping schedule, and, based on the information received from neighbors, they adjust their schedule so that they all sleep at the same time. During data periods, a node with data to send will contend for the medium (RTS-CTS exchange). If the node acquires the medium or if it has data to receive, it will not sleep in the next period and the data will be exchanged. After that, if there is still enough time in the sleep period, the node goes to sleep. If a node does not have data to transmit or receive, it will sleep.

There has been considerable research activity in power-aware MAC protocols. Besides S-MAC, another notable example is the T-MAC [23] protocol, a contention-based MAC for wireless sensor networks which uses an active/sleep duty cycle. TRAMA [16] is an example of a power-aware scheduled-based (time-slotted) MAC Protocol. It establishes transmission schedules in a way that: it is self adaptive to changes in traffic, node state, or connectivity.

3. Energy consumption instrumentation

The main goal of our instrumentation is to provide a "common ground" through which the effectiveness of different power aware techniques, at a specific layer of the protocol stack or across different layers, can be evaluated. Our model is simple and uses an approach similar to [2]. However, our instrumentation does account for time and energy consumed in all radio states, including low-power sleep mode.

Due to the shortcomings of the energy models available in existing simulation platforms (see Section 2 for details), developers of power-aware protocols often implement their own energy models (e.g., [16], [13]), usually at the layer of the protocol stack they are developing at. Another benefit of the proposed instrumentation is that, since it is implemented at the physical layer, it can be used by any protocol layer.

3.1. Description

The proposed energy model considers all possible radio operation modes, namely: **Transmitting**, i.e., radio is transmitting data; **Receiving**, i.e., radio is effectively receiving data; **Overhearing**, i.e., radio is receiving data that is not destined to the node[1]; **Idle**, i.e.,

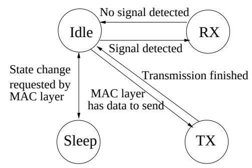


Figure 1. State diagram for radio modes

radio is ready to receive or transmit; **Sensing**, i.e., radio has detected some signal, but is not able to receive it; **Sleeping**, i.e., radio is in low power, and this is not able to receive or transmit. Note that *sensing* and *overhearing* states are a special case of the *receiving* state. Figure 1 shows the state diagram, which depicts the main radio states and how state transitions occur.

The power dissipated by the radio can be calculated using the expression $P = V * i$, where V and i are the voltage and current specific to the radio being used and are typically available from the radio data sheet. The time the radio spends in a certain state depends on the packet size and the transmission rate and is given by $t = PacketSize/TxRate$.

Thus, for each state, energy consumption is calculated as $E_y = \alpha_y * t_y$, where α_y represents the power dissipated by the radio while in state y, and t_y represents the time spent in state y. State y can be transmission, reception, idle, sleep or sensing. Note that the values of α_y are radio-specific.

3.2. Implementation

The energy model was implemented at the radio/physical layer of both GloMoSim and QualNet. The implementation includes: (1) the necessary physical layer infrastructure to account for all possible radio modes (as specified above), and (2) an interface between the physical- and MAC layers to control the radio modes (e.g., switch radio on/off, overhearing versus reception, etc.).

The physical layer support for the energy consumption instrumentation includes: (1) the addition of the SLEEP state, (2) addition of a data structure for the energy model, (3) and implementation of energy consumption accounting functions. The radio/physical implementations were modified such that these functions are called every time a radio state change occurs.

Since the radio layer now supports the "sleep" state, it is necessary to provide functions for the MAC layer to set the radio state to and from sleep mode. Functions $GlomoEnergyRadioWakeUp$ and $GlomoEnergyRadioGoToSleep$ provide this functionality. Another aspect where the need of interaction between MAC- and physical layers become clear is in order to identify if a received packet was in fact received or overheard (i.e., the destination for the packet is not the current node). The energy model assumes that all packets received are overheard, and thus the function $GlomoEnergyUpdateEnergyRx$ should be used every time a received packet is destined to the node. If this function is not used, all energy due to receiving will be accounted as overhearing.

Each time the radio changes state, energy consumption information is updated. In order to support power-aware protocols, current energy spent information in a node can be obtained by using the function $GlomoGetCurrentEnergySpent$. For example, a power-aware routing protocol may want to examine the current energy information at nodes to compute paths that only include nodes whose energy are above a certain threshold.

Through a configuration file, the user defines the energy consumption parameters. Statistics provided by the energy model include: total energy consumption, energy consumption per state, time spent in each state (including or not a "warm up" period).

4. Validation

We validate the energy consumption instrumentation analytically and through simulations. For the latter, we compare QualNet/GloMoSim instrumented with our model against the original QualNet/GloMoSim. The protocol used is IEEE 802.11 DCF, which uses transmitting, receiving and idle modes. Since the idle state typically consumes the same energy as receiving, we can directly compare our model to GloMoSim/QualNet's which only considers transmitting and receiving states.

The other validation step is to compare analytically the results obtained when a simple topology is used. We performed this step for both IEEE 802.11 DCF and S-MAC. Finally, we validate the model by reproducing experiments conducted on a testbed implementation of S-MAC [26] and comparing the results from the testbed with our simulations.

We then showcase the use of our model by comparing, through simulations, S-MAC's power efficiency against that of 802.11, and AODV against DSR. We show that, through the use of an adequate energy instrumentation, it is possible to get insight into where

1 For deciding whether the radio is receiving or overhearing, the energy model needs information from the MAC layer regarding whether the node is the recipient of the frame.

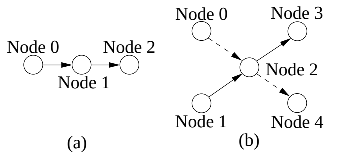


Figure 2. (a) Three-node topology and (b) Five-node topology used in validation.

energy is spent and how energy efficiency can be improved. All simulation results presented are averaged over 10 runs with different seeds.

4.1. Instrumented versus original GloMoSim/QualNet

In this experiment, the goal is to validate our model by comparing the energy consumed by IEEE 802.11 DCF using original GloMoSim/QualNet against results obtained using GloMoSim/QualNet with our instrumentation.

We used the default values for all parameters in the configuration file, i.e., the transmission rate is set at 11 Mbps, and the power consumption is 900 mW for both receiving/idle and transmitting states. The transmission range for each node is 100m (receiver threshold is -75dB).

The scenario used is a string topology with 3 nodes (two-hop topology), as shown in Figure 2(a). All nodes are stationary and routes are static. CBR traffic is generated from node 0 to 2 40 times with 5 second interval; the data size is 200 bytes. A simulation run lasts 250 seconds.

Table 1 shows the total energy consumed by each node for both original- and instrumented GloMoSim/QualNet. Note that IEEE 802.11 DCF only uses transmitting, receiving and idle modes, where consumption in idle state is typically the same as in receiving. Therefore, we can directly compare our model to GloMoSim/QualNet's, which only considers transmitting and receiving states. Thus, for this particular case, the values obtained are exactly the same.

	Original	Instrumented
Node 0	224999.46	224999.46
Node 1	224998.74	224998.74
Node 2	224999.28	224999.28

Table 1. Total energy consumption in original versus instrumented GloMoSim/QualNet.

4.2. Analytical versus simulation results

For this validation step, we use the specifications for the TR1000 [18] radio, which is designed for short-range wireless data communication, supports transmission rates of up to 115.2 Kbps, and has the sleep state built in. Power consumption is 13.5 mW, 24.75 mW and 15 μW in receiving/idle, transmitting, and sleeping states, respectively. The transmission range for each node is set to 100m (receiver threshold is -75dB). Data rate is 19.2 Kbps. Packet sizes are 20 bytes for RTS, 14 bytes for CTS and ACK, 380 bytes for DATA, and 24 bytes for SYNC.

The topology used is composed of five nodes as shown in Figure 2(b). Nodes 0 and 1 are sources, 3 and 4 are sinks, and 2 must route all the traffic in this two-hop network. The five-node topology was chosen in order to provide a scenario, which includes the possibility of collisions, overhearing and sensing. Static routes were set, so there is no influence from routing protocols. The experiment simulates 3 seconds of real time, which is the time needed to transmit one packet from sources to destinations.

802.11 Considering that each source is going to transmit one data packet, Table 2 shows the transmitted and received (and overheard) packets per node when 802.11 is used.

Node	Transmitted	Received	Overhears
0	RTS+DATA	CTS+ACK	(CTS+ACK) + 2*(RTS+DATA)
1	RTS+DATA	CTS + ACK	(CTS+ACK) + 2*(RTS+DATA)
2	2*(CTS+ACK) + 2*(RTS+DATA)	2*(CTS+ACK) + 2*(RTS+DATA)	
3	CTS+ACK	RTS+DATA	(RTS+DATA) + 2* (CTS + ACK)
4	CTS+ACK	RTS+DATA	(RTS+DATA) + 2* (CTS + ACK)

Table 2. Packets transmitted and received per node for 802.11.

Based on the information on Table 2, data rate and packet size, we can calculate the time each node spent in each state, shown in Table 3. Note that the time corresponding to receiving state is split between receiving and overhearing. The time the radio is not transmitting or receiving is spent in idle state. Also IEEE802.11 doesn't support sleep state and thus no time should be spent in it, and thus it is omitted in the table.

To obtain the corresponding simulation results, we use QualNet and send one packet from each source node to the corresponding sink node according to the topology in Figure 2(b) with 802.11 as the underlying MAC protocol. Table 4 presents time spent in each state, i.e., TX, RX, overhearing, sensing, and idle.

Node	TX	RX	Overhearing	Idle
0	0.166	0.011	0.344	2.479
1	0.166	0.011	0.344	2.479
2	0.356	0.369	0.000	2.275
3	0.011	0.166	0.189	2.634
4	0.011	0.166	0.189	2.634

Table 3. Analytical model: time spent in each relevant state for 802.11.

Node	TX	RX	Over-hearing	Sensing	Idle	Sensing + Idle
0	0.1787	0.0121	0.3695	0.2028	2.2369	2.4397
1	0.1787	0.0121	0.3695	0.2028	2.2369	2.4397
2	0.3816	0.3815	0.0000	0.0000	2.2369	2.2369
3	0.0121	0.1787	0.2028	0.3695	2.2369	2.6064
4	0.0121	0.1787	0.2028	0.3695	2.2369	2.6064

Table 4. Simulation: time spent in each relevant state for 802.11.

The small difference between analytical and simulation results for TX, RX and overhearing is due radio synchronization and internal delays intrinsic to the simulation. For each packet to be transmitted, the simulator will add the synchronization time and delays to the transmission time, and thus it propagates to all results. The time spent in sensing is caused by nodes nearby, but not in range (radio is not able to lock the signal). For example, in the topology of Figure 2(b), node 0's sensing time is given by the sum of the transmission times of nodes 1, 3 and 4. Because the sensing range depends on the radio properties (e.g., transmission power of the neighbors, radio sensitivity for the receiving node, distance between nodes, etc), we decided not to consider this state in the analytical model. Thus when comparing simulation results to the analytical results, sensing time should be added to idle time, resulting in values presented in the last column of Table 4.

S-MAC Next, we repeat the same exercise using S-MAC as the underlying MAC protocol. By using S-MAC, other factors must be taken into account. Besides the time spent in transmitting and receiving data, it is necessary to account also for the transmission of SYNC frames[2]. S-MAC makes use of low-power sleep state by switching nodes to sleep if a CTS, DATA, or ACK from another node is received. Table 5 summarizes the packets exchange for S-MAC.

Considering packets exchange in Table 5, data rate and packet size, we can calculate the time each node spent in each state, shown in Table 6. Note that the time corresponding to receiving state is split between receiving and overhearing. In order to compute the time

Node	Transmitted	Received	Overhears
0	RTS+DATA + SYNC	CTS+ACK + SYNC	CTS + 2*(RTS+DATA)
1	RTS+DATA + SYNC	CTS + ACK + SYNC	CTS + 2*(RTS+DATA)
2	2*(CTS+ACK) + 2*(RTS+DATA) + SYNC	2*(CTS+ACK) + 2*(RTS+DATA) + 4*SYNC	
3	CTS+ACK + SYNC	RTS+DATA + SYNC	(RTS+DATA) + 2*CTS
4	CTS+ACK + SYNC	RTS+DATA + SYNC	(RTS+DATA) + 2*CTS

Table 5. Packets transmitted and received per node for S-MAC.

spent in idle state, we calculate how many listen periods fit within the 3-second simulation runs; from that, we subtract the time spent transmitting and receiving SYNCs, RTSs and CTSs. Similarly, we can estimate the time spent in sleep state by calculating how many sleep periods fit within a simulation run, and from that subtract the time spent transmitting and receiving DATA and ACKs. Note that ideally no DATA should be overheard, because the data portion of the listen period is long enough to accommodate RTS and CTS packets.

Node	TX	RX	Overhearing	Idle	Sleep
0	0.1748	0.0217	0.3342	0.6335	1.8359
1	0.1748	0.0217	0.3342	0.6335	1.8359
2	0.3617	0.3917	0.0000	0.5685	1.6781
3	0.0217	0.1742	0.1758	0.6335	1.9948
4	0.0217	0.1742	0.1758	0.6335	1.9948

Table 6. Analytical model: time spent in each relevant state for S-MAC.

To get simulation results, we again use QualNet to send one packet from each source node to the corresponding sink node in the topology of Figure 2(b) using S-MAC as the underlying MAC. Simulations run for 3 seconds after the warmup period [3]. Table 7 presents time spent in each state as tracked by the energy model.

Node	TX	RX	Over -hearing	Sensing	Idle	Sleep
0	0.1842	0.0508	0.3381	0.0620	0.4592	1.9057
1	0.1811	0.0508	0.3381	0.0650	0.4592	1.9057
2	0.3974	0.4147	0.0000	0.0000	0.4592	1.7287
3	0.0262	0.2108	0.1695	0.0514	0.4592	2.0828
4	0.0242	0.2108	0.1685	0.0545	0.4592	2.0828

Table 7. Simulation: time spent in each relevant state for S-MAC.

Similarly to the results obtained for 802.11, we observe a difference (10% on average) between analyti-

2 Nodes periodically exchange SYNC frames in order to identify their one-hop neighbors and define their schedule.

3 S-MAC needs a 20 to 30-second warm-up period for initial neighbor synchronization

cal and simulation results due to radio synchronization and internal delays intrinsic to the simulator. Note that the calculation for sensing time in S-MAC is not as simple as for IEEE 802.11, since S-MAC has a sleep state. But still in order to compare analytical and simulation results, sensing and idle time should be added: 0.5212 for node 0, 0.5242 for node 1, 0.4592 for node 2, 0.5106 for node 3 and 0.5137 for node 4.

4.3. S-MAC testbed versus simulation

We also validate the proposed energy model by comparing results obtained from an implementation of S-MAC on a real sensor network testbed [26] against simulations that tried to reproduce the testbed experiments. Since details of the testbed experiment were not available, an absolute comparison is not possible. Our goal is thus to obtain a qualitative evaluation of the model. Once again, we used the TR1000 [18] radio parameters as described above. The same five-node topology is used and is shown in Figure 2(b). Nodes 0 and 1 are sources, 3 and 4 are sinks, and 2 must route all the traffic in this two-hop network. CBR traffic with packet size of 380 bytes is sent throughout the whole simulation period. Simulation time varies with inter-arrival period in order to keep the number of packets transmitted constant for all scenarios.

The graph in Figure 3(a) shows the average energy consumed for source and intermediate nodes for each different packet inter-arrival time. When compared to the energy consumption graphs presented in [26] (Figures 8 and 10), we observe similar behavior for all nodes. The intermediate node spends more energy because it needs to forward data and thus cannot sleep as much as the other nodes. Also, the energy consumption increases because the simulation time (or the data collection time in [26]) increases with the inter-arrival period.

5. Protocol comparison

The goal of these experiments is to showcase how our instrumentation can help evaluate power-aware protocols, irrespective of their position in the protocol stack. In the first example, we evaluate power-aware MAC protocol, namely S-MAC. In the second example, we evaluate two well-known routing protocols for multihop ad-hoc networks (MANETs), namely AODV [15] and DSR [12].

5.1. MAC protocol

In this specific example, we evaluate power-aware MAC protocol, namely S-MAC. For these experiments, we employ a scenario including fifty nodes uniformly distributed over 1000x1000m^2. The parameters in the configuration file were again set according to the TR1000 [18] radio specifications. A CBR application with 10 sources and 10 sinks sending 380-byte packets was used. AODV [15] was used as the routing protocol.

The graph in Figure 3(b) shows the average energy consumed per node for different packet inter-arrival times. Note that the different inter-arrival periods do not seem to affect the average time spent in each state because results are averaged over all nodes in the area (which might include nodes that do not transmit or receive at all).

Besides providing the overall energy consumption, our instrumentation allows a better understanding of which radio states are predominant and thus how energy savings can be achieved. Table 8 shows the time spent in each radio state (TX, RX, idle, sleep, sensing and overhearing) for message inter-arrival period of 1sec. These results demonstrate that the ability to use low-power sleep state yields considerable energy savings.

Protocol	TX	RX	Overhearing	Sensing	Idle	Sleep
802.11	0.92	0.68	0.01	10.09	137.20	0.0
S-MAC	0.94	0.63	0.03	22.49	29.71	68.38

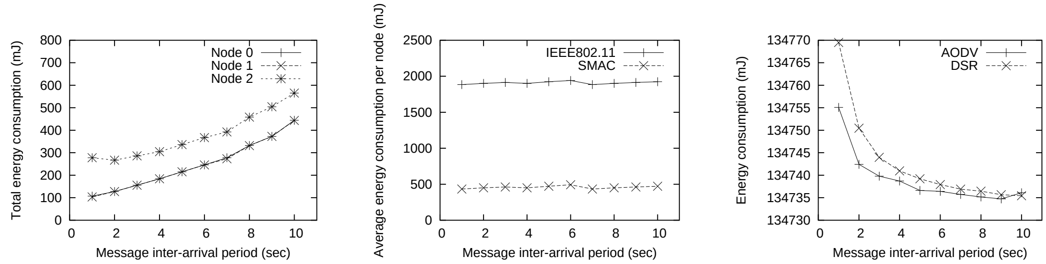
Table 8. Time spent in each state for 1sec message inter-arrival time.

Since the instrumentation not only gives the time, but also the energy consumed in each radio state, it provides a better understanding of the protocols under consideration. For instance, in [16], the energy savings are attributed to the length of the sleep time.

5.2. Routing Protocols

In order to show the ability of the instrumentation to account for energy consumption irrespective of the stack layer we want to evaluate, we compare energy consumption for two MANET routing protocols: AODV [15] and DSR [12].

For these experiments, we employ a scenario including fifty nodes uniformly distributed over 500x500m^2. The parameters in the configuration file were set according to the WaveLAN radio [7], and thus the power consumption is 900mW for both receiving and idle mode, and 1400mW for transmitting mode. The transmission rate is 11Mpbs. A CBR application with 10 sources and 10 sinks sending 380-byte packets was used. Simulation runs for 150 seconds.



(a) S-MAC energy consumption.

(b) Simulation results for S-MAC and 802.11.

(c) Simulation results for AODV and DSR.

Figure 3. Simulations results using instrumented GloMoSim/QualNet.

The graph in Figure 3(c) shows the average energy consumed per node for different packet inter-arrival times. Since the simulation had the same duration for all different inter-arrival times, the scenarios with smaller inter-arrival times were able to deliver more data packets, thus spending more time in transmission mode, and therefore had larger average energy consumption. In order to understand why DSR consumes slightly more energy per node than AODV, it is useful to look at the time spent in each state. Table 9 shows the time spent in each state for inter-arrival time of one second. In this case, DSR spends about 60% more time in transmission, receiving, overhearing and sensing states than AODV. This is due to the fact that DSR packets are longer, since they carry source routes.

Protocol	TX	RX	Over -hearing	Sensing	Idle	Sleep
AODV	0.0491	0.0463	0.2214	1.8519	147.5320	0.0000
DSR	0.0779	0.0737	0.3643	2.9891	146.1956	0.0000

Table 9. Time spent in each state for 1sec message inter-arrival time.

6. Conclusions and Future Work

This paper presented our work on instrumenting network simulators to enable them to adequately and accurately account for the energy consumed by ad hoc network protocols' communication-related tasks. This is accomplished by explicitly accounting for all possible radio states, i.e, transmitting, receiving, overhearing, idle, sensing, and sleeping, and considering the different energy costs associated with each of them. Another contribution of our energy consumption instrumentation is to allow the energy accounting to be done automatically by the simulator irrespective of what layer of the stack the protocol designer is working. For example, simulation results obtained using instrumented QualNet with our energy consumption accounting are used to validate the analytical model proposed to evaluate energy consumption in IEEE 802.11 single-hop wireless networks [3].

The instrumentation energy model is validated analytically and through simulations using two MAC protocols, i.e., IEEE 802.11 DCF and S-MAC. We also showcase our instrumentation's ability to evaluate energy consumption of protocols by comparing S-MAC against 802.11, and AODV against DSR.

As future work, we plan to extend our energy consumption instrumentation to real testbeds, e.g., Berkeley Motes [24]. By doing so, we will be able to validate the results obtained through simulations.

In low-power sensor networks, communication is known as one of the major sources of energy consumption. However, when sensor networks include more sophisticated sensors such as cameras, energy consumption due to processing and sensing are no longer negligible. We will investigate processing models in order to understand the energy consumption trade-offs between communication and processing/sensing. This will allow us to achieve an effective balance between energy efficiency and data fidelity. Also, we will be able to extend the energy consumption instrumentation to include processing and sensing. This will likely become a valuable tool for power-aware protocol designers.

Acknowledgment

The authors would like to thank Venkatesh Rajendran for providing the S-MAC implementation for QualNet, Roberto Manduchi for his valuable comments, and the anonymous reviewers for their suggestions to improve this paper.

References

[1] M. Bhardwaj and A. P. Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignments. In *Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.

[2] J.-C. Cano and P. Manzoni. A performance comparison of energy consumption for mobile ad hoc network routing protocols. In *8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, San Francisco, CA, September 2000. IEEE Computer Society.

[3] M. M. Carvalho, C. B. Margi, K. Obraczka, and J. J. Garcia-Luna-Aceves. Modeling energy consumption in single-hop ieee 802.11 ad hoc networks. In *Thirteenth International Conference on Computer Communications and Networks (ICCCN'04)*, Chicago, USA, October 2004.

[4] B. Crow, I. Widjaja, J. Kim, and P. Sakai. Ieee 802.11 wireless local area networks. *IEEE Communications Magazine*, 35(9):116–26, September 1997.

[5] E. J. Duarte-Melo and M. Liu. Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks. In *IEEE Globecom*, Taipei, Taiwan, November 2002.

[6] J. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz. Measurement and simulation of the energy consumption of an WLAN interface. Technical Report TKN-02-010, Technical University Berlin, Telecommunication Networks Group, Germany, June 2002.

[7] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies.INFOCOM 2001*, volume 3, pages 1548–1557. IEEE, April 2001.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *33rd Hawaii International Conference on System Sciences (HICSS '00)*, Hawaii, January 2000.

[9] W. B. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts Institute of Technology, 2000.

[10] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba. ACPI: Advanced configuration and power interface. http://www.acpi.info/, 2004.

[11] ISI/USC. ns-2. http://www.isi.edu/nsnam/ns/, 2003.

[12] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[13] D. Kim, J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano, and P. Manzoni. Performance analysis of power-aware route selection protocols in mobile ad hoc networks. In *IEEE Networks 2002*, Atlanta, GA, August 2002.

[14] E. Lochin, A. Fladenmuller, J.-Y. Moulin, and S. Fdida. Energy consumption models for ad-hoc mobile terminals. In *Med-Hoc Net*, 2003.

[15] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing. http://www.faqs.org/rfcs/rfc3561.html, July 2003.

[16] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *ACM SenSys 03*, Los Angeles, CA, November 2003.

[17] J. M. Reason and J. M. Rabaey. A study of energy consumption and reliability in a multi-hop sensor network. *Mobile Computing and Communications Review*, 8(1):84–97, 2004.

[18] RFM. TR1000 datasheet. http://www.rfm.com/products/data/tr1000.pdf, 2003.

[19] SCT. QualNet user manual. http://www.scalable-networks.com/, 2003.

[20] H. Singh and S. Singh. Energy consumption of tcp reno, newreno, and sack in multi-hop wireless networks. In *SIGMETRICS02*, 2002.

[21] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in handheld devices. *IEICE Trans. on Communications*, 8(E80-B):1125–1131, 1997.

[22] F. Tari, P. Rong, and M. Pedram. An energy-aware simulation model and a transaction protocol for dynamic workload distribution in mobile ad hoc networks. In *Proceedings of the 21st International Conference on Computer Design: VLSI in Computers and Processors*, San Jose, CA, October 2003.

[23] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *ACM SenSys 03*, Los Angeles, CA, November 2003.

[24] A. Woo. Mote documentation and development information. http://www.eecs.berkeley.edu/ awoo/smartdust/, 2000.

[25] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin. Topology control protocols to conserve energy in wireless ad hoc networks. Technical Report CENS Technical Report 6, Center for Embedded Networked Sensing, USA, January 2003.

[26] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.

[27] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a library for parallel simulation of large-scale wireless networks. In *12th Workshop on Parallel and Distributed Simulations – PADS '98*, Banff, Alberta, Canada, May 1998.