Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks

Ya Xu, John Heidemann, Deborah Estrin
USC/ISI Research Report 527

October 12, 2000

Abstract

In this paper, we present two algorithms for routing in energy-constrained, ad hoc, wireless networks. Nodes running our algorithms can trade off energy dissipation and data delivery quality according to application requirements. Our algorithms work above existing on-demand ad hoc routing protocols, such as AODV and DSR, without modification to the underlying routing protocols. Our major contributions are: algorithms that *turn off* the radio to reduce energy consumption with the involvement of *application-level information*, and the additional *use of node deployment density* to adaptively adjust routing fidelity to extend network lifetime. Algorithm analysis and simulation studies show that our energy-conserving algorithms can consume as little as 50% of the energy of an unmodified ad hoc routing protocol. Moreover, simulations of adaptive fidelity suggest that greater node density can be used to increase network lifetime; in one example a four-fold increase in density doubles network lifetime.

1 Introduction

Multihop ad hoc networking has been the focus of many recent research and development efforts. It has wide application in military, commercial, and educational environments such as wireless office LAN connections, mobile phones, PDAs, or computers, and sensor networks.

Current studies of ad hoc network routing protocols have focused primarily on protocol design and evaluation in terms of routing packet overheads and loss rates. The ad hoc network typically consists of energy limited nodes, so designing power-conserving protocols deserves study. For some scenarios, power is *the* important metric. For example, in sensor networks, nodes are casually placed and remain unattended for long periods of time, sensing and reporting objects until sensor nodes run out of power.

We examined existing ad hoc routing protocols using power consumption models with physical radio simulation provided by [3]. The physical radio characteristics approximates the Lucent WaveLAN direct sequence spread spectrum radio using the IEEE 802.11-1997 protocol. We found that significant energy is consumed even when there is no traffic in the network. The reason is that idle state energy dissipation dominates the total energy dissipation in an IEEE 802.11 network. Simple optimizations in ad hoc routing (to reduce message transmission) is therefore unlikely to substantially improve energy usage. This observation motivates the two primary contributions of our paper:

1. Algorithms that *turn off* the radio to improve power consumption with the involvement of *application-level information*: The cost of turning off the radio is added latency and possibly more packet loss compared to unmodified protocols. Therefore we must design our energy saving algorithm to find a trade-off between energy conservation and data-delivery quality. We argue that application level information is needed to make the best tradeoff. The type of application level information needed is quite simple; for example, whether a node needs to send or forward data.

2. The *use of node deployment density* to adaptively adjust routing fidelity: This contribution is based on the observation that in ad hoc networks where nodes are densely deployed (many

can hear each other), some nodes are interchangeable for routing purposes. We show how to use this information to further increase node duty cycles and to extend the lifetime of the network as a whole. This approach is one example of *adaptive fidelity* [8], in this case keeping the fidelity of network reachability constant while adapting node behavior to extend network lifetime.

The concept of powering down radios to save power has been introduced elsewhere. PAMAS presents a MAC protocol that conserves energy by turning off radios to avoid overhearing cross-traffic [20], and TDMA-based protocols can conserve energy with low duty cycles (for example, in WINSng [16]). Unlike this prior work, we are able to turn the radio off for much longer periods by using information at the routing and application layers. PicoNet nodes are designed with very long duty cycles [2]; however, their approach applies to fixed node and base station architectures whereas our approach extends existing ad hoc routing techniques. We examine related work more thoroughly in Section 5.

Simulation studies shows that our energy-conserving algorithms will consume as little as 50% of the energy of an unmodified ad hoc routing protocol over the same duration. In networks with a fixed energy budget, typical networks all run out of power at the same time, while our adaptive fidelity approaches keep half of the network alive 50% longer, and some nodes alive twice as long. We present these results in detail in Section 4.

2 Motivation

A number of routing protocols have been proposed to provide multi-hop communication in wireless, ad hoc networks [14, 4, 15, 13]. Traditionally these protocol are evaluated in terms of packet loss rates, routing message overhead, and route length [3, 11, 7]. Since ad hoc networks will often be deployed using battery-powered nodes, comparison and optimization of protocol energy consumption is also important (as suggested for future work by some researchers [11]).

We first tried to understand whether these ad hoc routing protocols have different energy dissipation rates. We studied the AODV, DSR, DSDV, and TORA protocols using the ns-2 simulator [1]. Ns-2 provides Internet and wired network simulations and incorporates implementations of these ad hoc routing protocols contributed by CMU [3]. To this base we have added a revised implementation of AODV [7] and a model of communications energy consumption. We began with a simple energy model where packet reception consumes fixed energy, transmission consumes twice as much, and listening (but not receiving) is free. This model has been the basis for some prior studies of flooding [10].

We compared these protocols (AODV, DSR, DSDV, and TORA) with this simple energy model by measuring the energy consumed over a fixed duration. We employed a simple traffic model (described later in Section 4.2) and set energy for all nodes high enough that none were exhausted over the simulation lifetime. This study confirms that on-demand protocols such as AODV and DSR use less energy than a priori protocols (DSDV and TORA) while providing the same delivery quality (as observed before [3, 11]). On demand protocols do not do any routing when there is no traffic in the network. With our simple energy model this approach consumes no energy when idle. A priori protocols are constantly consuming energy by pre-computing routes even when no data will be sent. (Although TORA is described as an on-demand routing protocol, it depends on IMEP's periodic control messages [6]. We therefore do not treat TORA as on demand routing protocol.)

In other words, on demand protocols, by their very nature, are efficient in the energy consumed by routing overhead packets. As a result, energy use is dominated by routing protocol overhead. In fact, the major source of extraneous energy consumption was from *overhearing*. Radios have a relatively large broadcast range. All nodes in that range must receive each packet to determine if it is to be forwarded or received locally. Although most of these packets are immediately discarded, they consume energy with this simple radio model. This observation motivates approaches that avoid overhearing. The PAMAS protocol suggests a MAC-layer approach to minimize this cost [20]; TDMA protocols would also be applicable (for example [16]).

However, actual radios consume power not only when sending and receiving, but also when *listening* (the radio electronics must be powered and decoding to detect the presence of an incoming packet). We adopted a more accurate energy model based on measurements of WaveLAN radios [21] where idle:receive:send ratios are 1:1.05:1.4. With this model, node idle time dominates energy consumption and all ad hoc routing protocols show similar energy consumption (within a few percent) in light or moderate traffic scenarios.

These studies, which were based on a more accurate power model, suggest that energy optimizations must actually turn off the radio, not simply reduce packet transmission. We therefore explore nodes that power down their radios much of the time. This approach is similar to the use of TDMA for power savings [16], or PAMAS [19]. However, unlike these approaches, we employ information from above the MAC-layer to control radio power. (We make use of the power management controls in IEEE 802.11 to control power.) The application- or routing-layers provide better information about when the radio is not needed.

We explore an additional optimization possible in ad hoc networks that are densely populated. We lengthen sleep intervals in this case since nodes are interchangeable for routing purposes. We show how to use this information to further increase node duty cycles and to extend the lifetime of the network as a whole. This approach is one example of *adaptive fidelity* [8], in this case keeping the fidelity of network reachability constant while adapting node behavior to maximize network lifetime.

More generally, we wish to introduce tuning knobs whereby a network designer can trade-off quality of data delivery for extended network lifetime.

3 Energy-conserving Routing Algorithms

In this section we present two application-driven energy-conserving ad hoc routing algorithms. These algorithms are based on the observations presented in Section 2: first, because radios commonly used for 802.11-like networks consume nearly as much power listening as receiving, the only way to substantially reduce energy consumption is by turning the radio off. Second, we can take advantage of information above the MAC-layer to control how long we can keep the radio turned off. Third, it is possible to take advantage of node density to further conserve power.

Our first algorithm is the *basic energy-conserving algorithm* (BECA). The basic idea is that nodes do not need to be listening and consuming power when they are not involved in sending, forwarding, or receiving data. The PAMAS protocol applies this result at the MAC level [19], turning off after determining packets are addressed elsewhere, but it still listens when idle to receive new packets. We improve this result using higher-level information to turn off the radio more frequently and for a longer duration, thus reducing the substantial energy dissipated during the idle state. We describe this algorithm in detail in Section 3.1.

Our second algorithm is the *adaptive fidelity energy-conserving algorithm* (AFECA). This algorithm uses observations about node density to increase the time the radio is powered off. When many equivalent nodes are able to forward data, they power off for longer intervals. In a sense, AFECA adapts the number of nodes participating in ad hoc routing to keep a constant level of routing fidelity (number of nodes that will route packets) to reduce energy consumption. We describe this algorithm in Section 3.2.

In principle these algorithms can be applied as modifications to any ad hoc routing protocol. We study on-demand ad hoc routing protocols such as AODV and DSR, for two reasons. First, on-demand protocols have been shown to perform better (in terms of packet loss) than a priori ad hoc routing protocols [3, 11]. Second, a priori routing protocols depend on periodic message exchanges. Care must be taken to avoid synchronization problems if combining our algorithms with such routing protocols.

3.1 Basic Energy-Conserving Algorithm

The goal of BECA is to minimize energy consumption by keeping radios powered-off as much as possible, trading higher latency for reduced energy use. We will show that this algorithm will establish routes in all cases where a standard ad hoc routing protocol would, although it may introduce longer latency.

Preliminary algorithm: In BECA,nodes are in one of three states: *sleeping*, *listening*, *active*. A state diagram is shown in Figure 1.

Initially nodes start out in the *sleeping* state. When *sleeping* the radio is off, not consuming power. In this state they keep their radio turned off for time T_s, then transition to *listening*. If when a node is *sleeping*, it has data to send, it transitions to *active* and starts sending the data. (Although

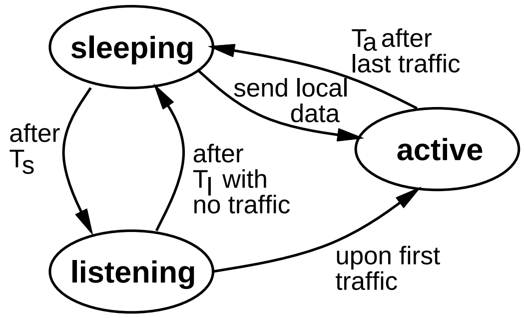


Figure 1: States in BECA

the radio is off, sensors or other parts of the node may be on.)

When in state *listening*, a node turns on its radio and listens for messages. It listens for time T_ℓ. During this time, if it gets a routing message and participates in the route, or if it decides to send data, it transitions to state *active*. Otherwise it returns to *sleeping* after T_ℓ.

When in the state *active*, a node sends or transits data. If at any time it hasn't sent or transited data in time T_a, it transitions to state *sleeping*.

We must control how this duty cycle interacts with ad hoc routing; sometimes the recipient of the routing request (RREQ) will be in sleep mode. We require that the ad hoc routing protocol retry requests every T_o seconds, and that it retry R times.

To manage interactions between BECA and the underlying ad hoc routing protocol we set T_ℓ to be the same as T_o, then we pick T_s as some multiple k of T_o and then adjust R to insure that some request will get through. Since we always listen for periods of T_o, we are assured that if another node is trying to establish a new route, we will hear their RREQ message sometime.

BECA and ad hoc routing: To illustrate this algorithm, in Figure 2 we consider routing. Initially we set T_s to $1 \times T_o$, so that nodes have a 50% duty cycle. When node A sends a routing request, node B is either sleeping or listening. If it is listening, the request is honored and the route is established; both nodes become *active* until data exchange is completed. In the figure, we assume the worst case, that B starts to sleep just as we send the initial RREQ. We are guaranteed that B will wake up in $T_s = T_o$ seconds and hear our second RREQ. On average, we add $T_o/2$ to latency.

We therefore conclude that for this value of T_s we can reduce energy usage by half, we will increase latency by at most T_s, and we can establish routes in $R = 2$ retries.

We can generalize this argument to multiple-hop routing. We observe that once the first hop hears the RREQ message, it transitions to *active*, and it stays in that state for at least T_a seconds. Since $T_a \geq T_o$, even RREQs that do not reach their final destination will keep intermediate nodes from sleeping. With $T_s = 1 \times T_o$, each hop will incur up to T_o additional latency in the worst case (we don't assume any synchronization between node sleep and wakeup patterns). Thus, for a network H hops in diameter, BECA adds at most HT_o to latency, on average $(HT_o)/2$, and we require $R = 2H$ retries to insure we succeed in establishing routes.

After a route is set up, those nodes that are not in the route will no longer receive RREQ message or data, so that they will return to *sleeping* (after T_a). Those nodes that are on the route will remain in *active* until data exchange ceases.

We can generalize this approach for $T_s = kT_o$ for $k > 1$ to get better duty cycles. Larger values of k linearly increase latency but improve power savings only by a factor of $1/k$. In Section 4 we examine the relationship between power savings, latency, and packet loss.

Accommodating packet loss: The above discussion has assumed that no routing requests are lost in transit. In real ad hoc networks packets can be lost because of data corruption, collisions, or node movement. To account for loss from the first two sources we transmit routing requests twice each listen or sleep interval, setting $T_\ell = T_s = 2 \times T_o$. To account for node movement we need to increase the number of retries.

3.2 Adaptive Fidelity Energy-Conserving Algorithm

In densely-populated ad hoc networks many nodes are interchangeable for routing purposes. Our Adaptive Fidelity Energy-Conserving Algorithm (AFECA) takes advantage of this observation to improve energy conservation by estimating node populating and increasing sleep time when other nodes are available. We will show that this approach increases network lifetime as node density increases.

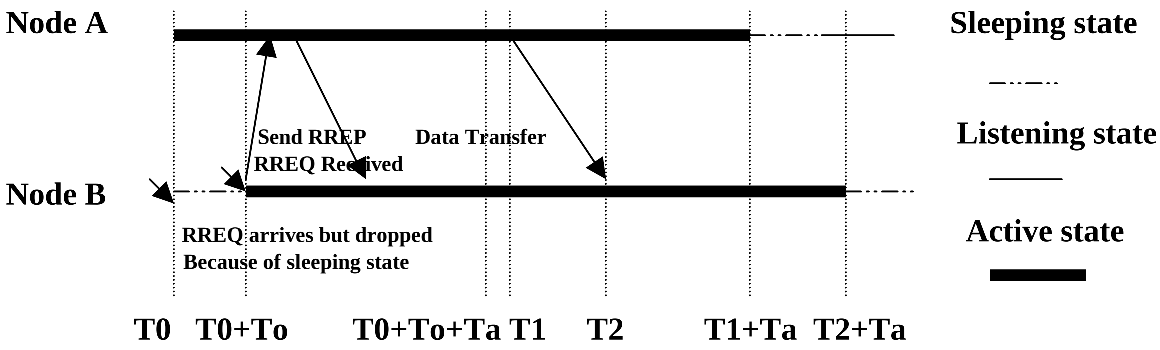


Figure 2: Establishing routing between to an adjacent (but possibly sleeping) node.

AFECA raises two design issues: how do we estimate neighborhood density, and, how do we adjust the duty cycle to account for this information.

In our prototype, each node estimates its neighborhood by keeping a list of what nodes it hears whenever it is listening. This list is treated as soft state, nodes are automatically removed from it if at any time it hasn't been updated in time T_e (we use a fixed T_e of 50 seconds in our experiments). By maintaining this list from information we happen to hear we avoid any additional message or energy overhead of explicit neighbor-discovery messages. We define the size of the neighborhood list as N.

In AFECA, each node increases its sleeping a some factor proportional to the number of nodes in its neighborhood. We define T_{SA} by node's actual sleeping time in order to differentiate it from BECA's T_s. In our prototype implementation, we define $T_{SA} = \text{Random}(1, N) \times T_s$ A node recomputes T_{SA} before it begins sleeping using it latest estimate of N. We expect an upper bound on T_{SA} is appropriate, although we do not currently set one.

We can define the aggregate duty cycle of N nodes using AFECA as the collective time they spend listening divided by the collective time they spend listening and sleeping. Assuming each node has the correct estimate of N, they will each listen for T_o and sleep for mean $N \times T_s/2$. For T_{SA}, therefore the aggregate duty cycle is $2/(2+N)$. We are currently considering other definitions of T_{SA}. We do not carefully try to insure accurate measurement of neighborhood size, primarily because there is a feedback effect in neighborhood estimation. If we underestimate it, neighboring nodes will spend more time awake and will be more likely to hear any neighbors and therefore increase their estimates. If we overestimate it, the converse holds. A secondary reason is that that incorrect estimates simply alter latency, not correctness.

This approach is what we use in our simulations. A more complete analysis of alternatives to estimate neighbor population and adjust T_{SA} are subjects of future work. We close with brief analysis of three cases that illustrate directions we are examining.

First, with completely passive neighbor discovery, nodes in a quiet network will forget about all neighbors. In this case AFECA simplifies to BECA with $T_s = T_o$ and has 50% duty cycle. When sensor networks are used for surveillance applications a quiet network (nothing detected) is the typical condition. In these cases, introduction of hard-state measures of neighborhood may be appropriate.

Second, our current definition of neighborhood works best when nodes are relatively evenly distributed. For extremely non-uniform topologies large latencies are possible. For example, consider an "H" topology (see Figure 3) where two rows of n nodes are on the vertical bars of the H, while a single node is on the horizontal bar. The center node will hear all nodes and assume a large neighborhood, but it is the only node that can hear both sides and so should rarely sleep. With our current approach, once the center node awakens it will remain awake as long as traffic passes through it (due to T_a), but initial latency will be high. Addressing this problem requires topological understand of the network. Fortunately, dense senor networks where AFECA is appropriate reduce the likelihood of extreme non-uniformity of distribution.

Finally, in restricted topologies we can compute optimal solutions to the problems of estimating

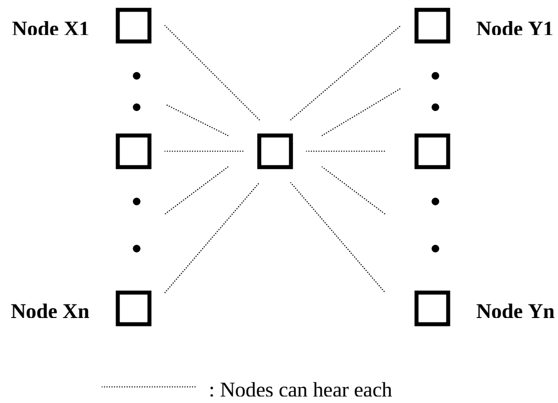


Figure 3: "H" topology: An example of extremely non-uniform topologies

neighborhood size and sleep duration. Consider an infinite line of nodes, each spaced 1 unit apart, numbered by the integers. With perfect radios that reach $1+\epsilon$, each node can hear only its neighbors and so T_s cannot be increased at all. Next assume the radio reaches $2+\epsilon$. Each node can hear 4 neighbors, so a duty cycle of one-third is appropriate. Using an argument similar to in the basic algorithm, nodes will always be able to get connectivity. In general, for this topology, neighborhood size $N = 2R$, and duty cycle is $1/(N-1)$.

4 Evaluation of Energy-Conserving Algorithms

Since our energy-conserving algorithms can only be studied analytically in restricted scenarios, we explore their performance with a set of simulation studies.

4.1 Simulation design

Our exploration of energy-conserving algorithms takes place in the ns-2 simulator [9]. We chose ns-2 because of our familiarity with it and its support for a wide range of ad hoc routing protocols, including DSR, AODV, DSDV, and TORA. Our work takes place in a snapshot of ns-2.1b5, which includes a modified and extended version of ad hoc routing contributed from CMU [17] and extended locally, and an improved AODV implementation from the AODV designers [7]. We have verified that our integration of the CMU code reproduces their results [3], and that our simulation results of unmodified ad hoc protocols are consistent with other published results [3, 7, 11].

To this base we have added a model of node energy consumption and prototypes of BECA and AFECA. Our energy model is based on results reported by Stemm et al. [21]. We assume that a radio consumes 1.15W when listening but idle, 1.2W receiving, and 1.6W sending. These values correspond to a 915MHz WaveLAN implementation of 802.11.

We have implemented BECA and AFECA as extensions to the AODV routing protocol [14]. The algorithms could be applied to other on-demand routing protocols. We plan to evaluate their effect on DSR in the future. We also plan to make our simulations available in future releases of ns.

In Section 3.1 defined BECA in terms of T_ℓ, T_s, and other constants. For our implementation of AODV, the maximum possible timeout value is 10s [7]; we therefore set $T_\ell = 10s$. The actual value of T_o varies, so it is difficult to compute the number of retries required to insure routes complete. Our implementation actually retries route requests endlessly, so we will not lose requests due to exhausting retries. We are considering modifying AODV to limit T_o to a smaller range so we can bound the number of retries.

AODV already times out routes automatically after 50–60s for intermediate or end nodes. We therefore set T_a at 60s, the larger of these values.

4.2 Experimental scenario

We placed 50 nodes in a square space (1500m by 1500m). Nodes move randomly using a way-point model. At each way point, a node pauses for a predefined time (600 seconds) and then moves to its next way-point at a randomly chosen speed uniformly distributed between 0 and 3m/s. This model does not attempt to reproduce a particular mobile networking scenario, but to provide conditions similar to those used in other studies of ad hoc routing [3, 11].

We generate traffic between these nodes by placing a number of constant-bit-rate (CBR) sources on nodes, randomly selecting sources and destinations. Each CBR source sends 512-byte packets for a random duration chosen uniformly between 0 and 1500 seconds. We place 25, 50, 75 or 100 sources (depending on load) and adjust their sending rate

protocol	traffic load (pkts/s) 5	10	15	20
AODV	0.04%	0.4%	0.3%	0.5%
BECA, $T_s = 2s$	0.13%	0.64%	0.21%	0.48%
BECA, $T_s = 3.3s$	0.09%	0.44%	0.42%	0.58%
BECA, $T_s = 5s$	0.27%	0.5%	0.22%	0.52%
BECA, $T_s = 10s$	0.12%	0.5%	0.4%	0.3%
BECA, $T_s = 20s$	4%	2.2%	1.1%	1.2%
BECA, $T_s = 30s$	1.6%	1.3%	1.2%	0.9%
BECA, $T_s = 40s$	2%	1.8%	1.3%	1.4%
BECA, $T_s = 50s$	6.5%	3.9%	3.1%	2.3%

Table 1: Comparison of BECA loss rates for different values of T_s.

between 1–10 packets/s to obtain aggregate traffic loads from 5–20 packets/s. We compute aggregate traffic load by averaging the sending rate of all nodes over the whole simulation.

We consider two levels of initial energy. First, we give all nodes an infinite amount of energy and vary algorithm parameters to compare loss rates and power consumption. Since nodes do not run out of power, these results are not complicated by node failure resulting network partitions. Using this model we evaluate choices of T_s and T_a for BECA. Second, we select the best choices for these parameters and network lifetime in a scenario where nodes have a limited amount of energy.

Finally, all graphs presented in this section represent the mean values from 10 simulation runs. Simulation runs vary traffic placement randomly, but all use the same movement patterns.

4.3 BECA performance evaluation

We first evaluate how BECA changes loss rates, latency, and energy consumption compared to unmodified AODV for cases where nodes do not run out of energy. We chose $T_\ell = 10s$, $T_a = 60s$, and vary T_s for a simulation lasting a fixed 1500s. Table 1 and Figure 4 shows how these metrics vary for a range of T_s values.

Loss rate: We evaluate loss rate by measuring the difference in the number of data packets sent vs. received. We calculate this as $(P_s - P_r)/P_s$ where P_s is the number of data packets generated by all traffic sources and P_r is the number of data packets delivered to all destinations.

Table 1 shows packet loss as a function of T_s. Loss rate of AODV and BECA for small values of T_s is very similar. Once a route is established unmodified AODV and AODV with BECA perform similarly. At large values of T_s, BECA observes higher loss due to packet losses while the route is established.

Routing latency: By causing nodes to sleep we add latency when setting up new routes. We measure routing latency as the time from the first routing request message until the routing reply is received.

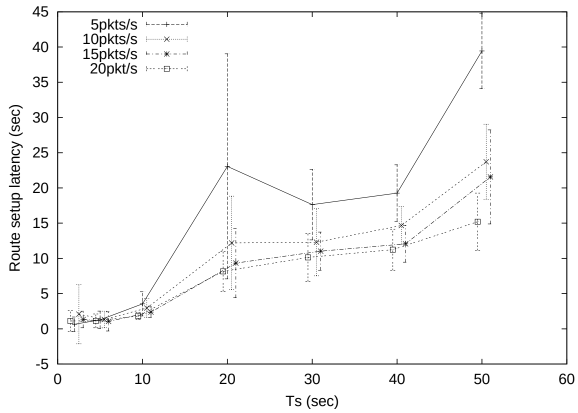
Figure 4(a) shows routing latency for BECA as T_s varies. For comparison, unmodified AODV has a fixed routing latency of 0.2s. Standard deviations are fairly high because delay of each routing request is between 0 and T_s, a wide range. The mean is not strictly monotonic because of statistical variation. (at low loads there are relatively few routing requests).

First, we observe that route latency grows roughly linearly with increasing T_s. Second, this growth is slightly lower at higher traffic rates. This effect is because in busier networks nodes are less likely to be sleeping. Finally, we conclude from this data that applications that use pass frequent, short messages cannot tolerate high values of T_s. Fortunately, short values of T_s achieve very good energy savings and have reasonable latency.

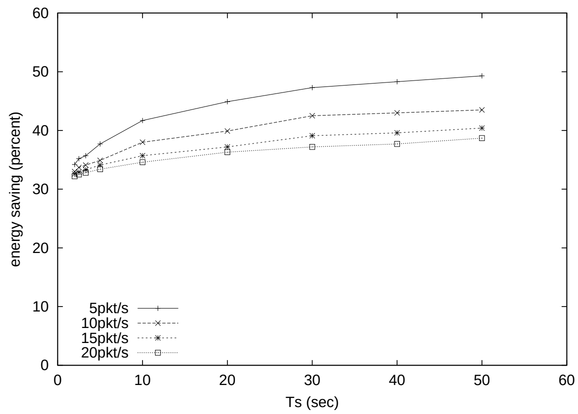
In addition to route latency, we also measured mean data packet latency. Mean packet latency for AODV and BECA at $T_s = 10s$ are both about 1%. Loss rates for data packets are similar because once the route is established, BECA keeps nodes on the route awake. Packets only suffer sleep-induced delay if the route changes.

Energy savings: Loss and latency are the costs of BECA; its benefit is energy savings. We compute the energy consumed in the simulation and compute how much lower this is than the same simulation with unmodified AODV. We calculate the percentage energy saved as $(E_r - E_s)/E_r$ where E_r is the total energy consumption for unmodified AODV and E_s is the energy consumption for BECA.

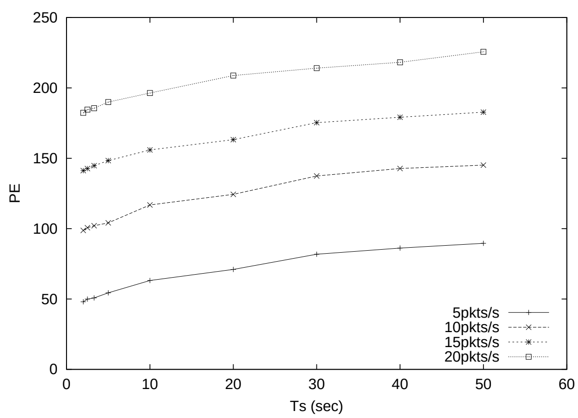
Figure 4(b) shows energy savings for various values of T_s. Standard deviations of energy consumption are very small (less than 1%) and so are not shown. Analysis predicted a 50% savings at $T_s = 10s$ (Section 3.1). Simulation validates shows



(a) Route setup latency



(b) Energy consumption



(c) delivery:energy ratio (PE)

Figure 4: Comparison of BECA to AODV for different values of T_s in terms of route setup latency, energy consumption, and PE.

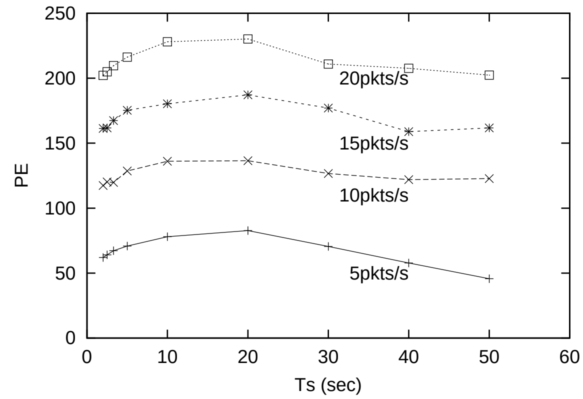


Figure 5: PE comparison of AFECA with different values of k for computing T_{SA}.

we reach 10% of this optimum at $T_s = 10s$ in our scenario.

We observe that there is less energy savings at higher traffic loads. More traffic leaves more nodes in active states, reducing time spent sleeping sleep time. However, even with the heaviest traffic of 20 packets/s, BECA still reduces energy consumption by 35% energy.

Selecting T_s: We also observe that much higher sleep times show no energy improvement. Based on the observations from Figure 4 we conclude that there is little to be gained from high T_s values.

To select an optimism T_s we need a metric that considers both packet loss and energy savings at the same time. We introduce the value PE to evaluate this ratio. PE $= P/E$, where P is the size of data delivered to the destinations in bytes, and E is the total energy consumed by all nodes of the network in Joules. Figure 4(c) shows this trade-off. Although PE increases monotonically for these scenarios, we consider a T_s of 10s to be reasonable, capturing the majority of efficiency while avoiding high route setup latency.

4.4 AFECA performance evaluation

AFECA defines node sleeping time as $T_{SA} = \text{Random}(1, N) \times k$. (see Section 3.2 for details). Our first task is to select k. Table 2 and Figure 5 compares the PE ratio for various values of k. A value of $k = 10s$ is best by this metric for our workload.

protocol	traffic load (pkts/s) 5	10	15	20
AODV	0.04%	0.4%	0.3%	0.5%
BECA, $T_s = 10s$	0.12%	0.5%	0.4%	0.3%
AFECA, $k = 10s$	0.45%	1.3%	0.97%	1.7%

Table 2: Comparison of loss rates for AODV, BECA ($T_s = 10s$), and AFECA ($k = 10s$).

We have suggested that PE is a measure of data transfer energy efficiency. Figure 5 shows that heavier traffic loads are correlated with higher PE ratios. At heavier traffic loads packets are delivered more energy-efficiently because intermediate nodes are able to forward data for multiple streams, thus reducing mean per-packet energy.

Figure 6 summarizes AFECA performance and compares the three protocols. Table 2 and 6(a) show that, as expected, BECA and AFECA are worse than AODV in terms of packet loss rate and route setup latency for these choices of parameters. In addition, AFECA delay is higher and shows more larger variance than BECA, as expected because of longer, random sleep times. However, the cost is bounded and modest.

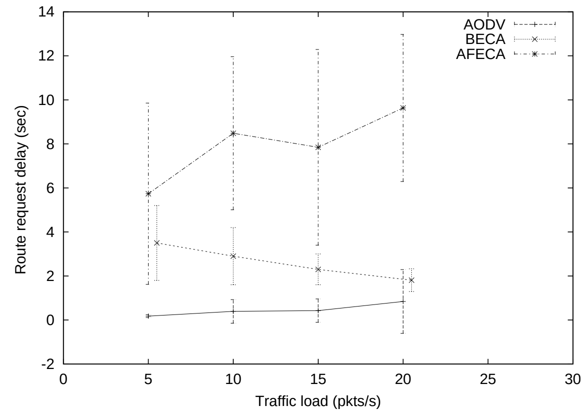
In Figure 6(b) we compare BECA and AFECA energy to unmodified AODV. With these parameters they reduce energy consumption by 35–45%, with AFECA 2–5% more thrifty than BECA. In Section 4.5 we will show that this savings has a noticeable effect over network lifetime. We also observe that higher loads offer less chance to save energy—more nodes must stay on to forward data.

Finally, Figure 6(c) compares the energy efficiency (measured as PE) for the protocols. AFECA aggressive power savings results in the consistently highest efficiency.

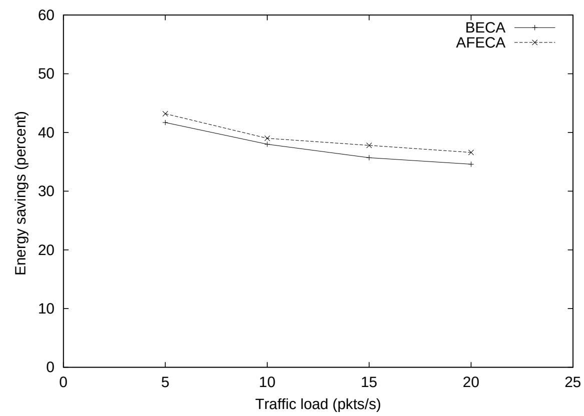
4.5 Evaluation with limited energy

Previous experiments have started each node with enough energy so that none run out during the simulation. We next examine simulations where nodes have limited energy to study the effect of nodes running out of power and leaving the network. We set the initial amount of energy for each node to 1000J. Nodes that send no packets and listen at all times will run out of power in 870s. We run simulations until all nodes are out of power.

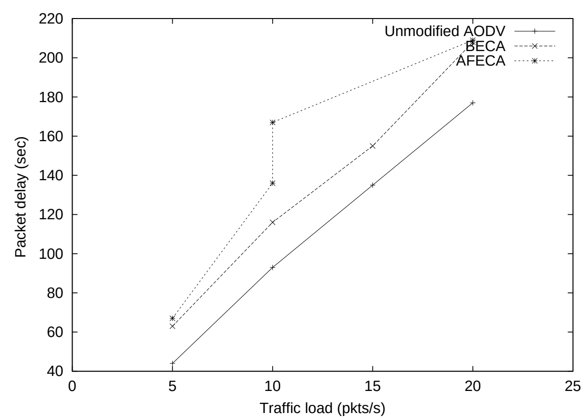
In this section we set $T_s = 10s$ according to the results from Section 4.3, and $k = 10s$ as described



(a) Route setup latency



(b) Energy consumption



(c) delivery:energy ratio (PE)

Figure 6: Comparisons of AODV, BECA ($T_s = 10s$), and AFECA ($k = 10s$).

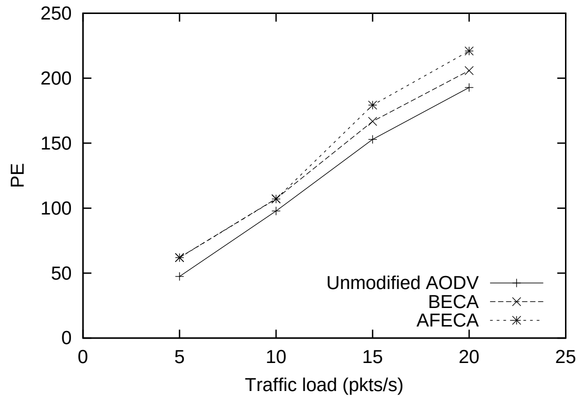


Figure 7: PE comparison of unmodified AODV, BECA, and AFECA under different traffic loads with limited energy.

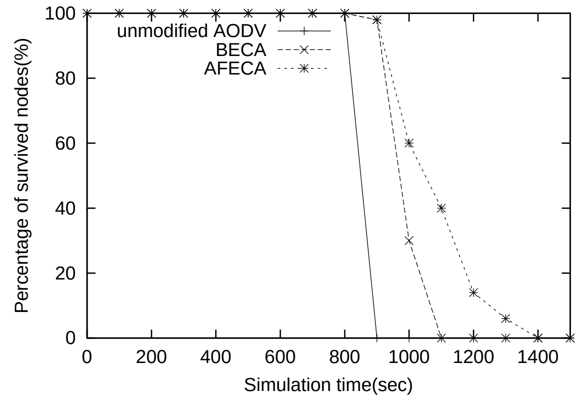


Figure 8: Network lifetime comparison of unmodified AODV, BECA, and AFECA under traffic load of 10 packets/s with limited energy.

in Section 4.4.

System efficiency: In Figure 7 we evaluate PE for unmodified AODV, BECA and AFECA with limited energy under different traffic load. Since this scenario runs all nodes out of power, the energy expended by all nodes is the same and this metric really measure the number of data packets each protocol is able to successfully deliver.

Both of the energy-conserving protocols are able to send more packets than unmodified AODV. At low loads they are equivalent, sending about 30% more data than AODV. At high loads AFECA rations energy better and sends up to 15% more data.

In Section 4.3 we argued that BECA and AFECA will show packet loss similar to AODV over short periods. This figure suggests that over longer time periods nodes will run out of energy. In this case, the energy-conserving nature of BECA and AFECA allows them to successfully deliver more packets than unomdified AODV.

Network lifetime: Our goal is to extend the lifetime of the network as a whole through energy conservation; our energy-conserving algorithms do this by putting nodes to sleep. Ultimately, the application wants to know how long the network can deliver information for it. It is difficult to directly evaluate this quantity directly, though, because application needs vary widely. We therefore measure node survival rates as a function of time, running the simulation until all nodes expire.

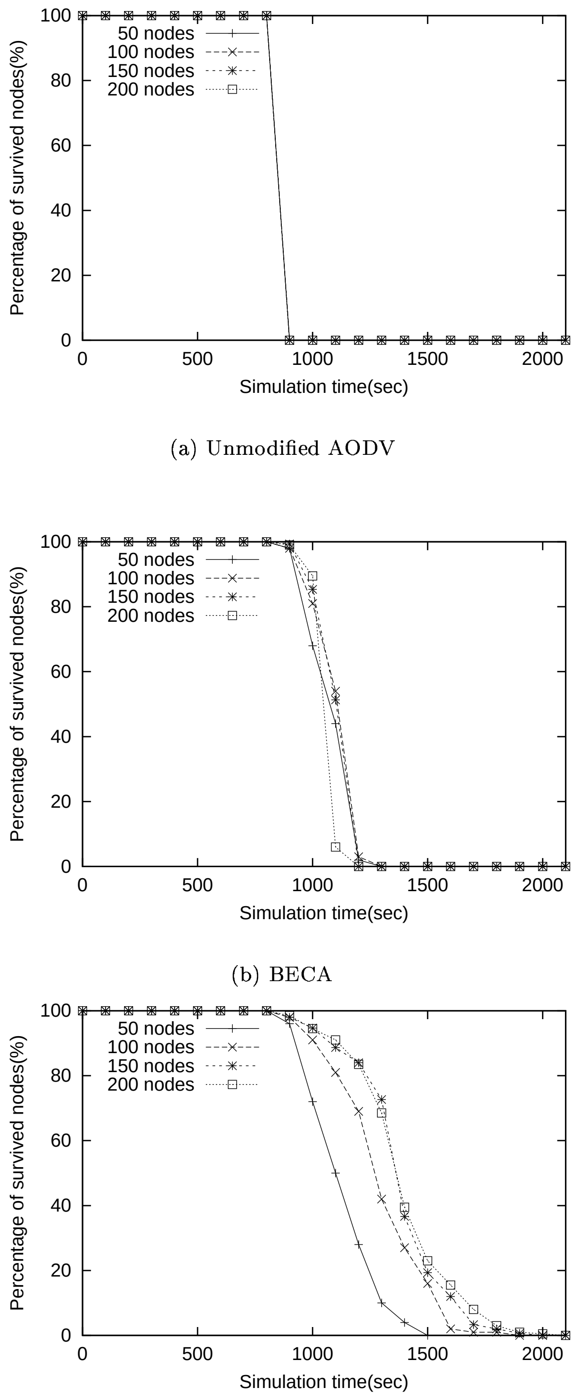
Figure 8 shows node survival as a function of time. (We consider each of our four traffic loads, but report only the 10 packet/s load since the other results are similar.) A first observation is that all AODV nodes run out of power at about the same time (870s into this simulation). This time is not affected by the network traffic load, confirming our claim that energy consumption in this scenario is dominated by idle-time consumption and independent of load.

By powering down radios, both BECA and AFECA networks last much longer than AODV. BECA is about 20% longer and AFECA about about 55% longer. These results bolster the argument that the reason BECA and AFECA show greater efficiency in Figure 7 is longer network lifetime and so more packet delivery.

Node density and lifetime: Our adaptive fidelity is designed to perform better in densely deployed networks. To evaluate this claim we considered a denser scenario: we place 50, 100, 150, and 200 nodes in a 1000m square area.

Figure 9 summarizes these results for the protocols and a traffic load of 10 packets/s. (Again, we looked at our other traffic loads and found similar results.)

From this data we conclude that AFECA is effective at making use of additional nodes to extend network lifetime. AODV and BECA performance is identical or about the same as node density increases, but a four-fold increase in density doubles network lifetime with AFECA.



(a) Unmodified AODV

(b) BECA

(c) AFECA

Figure 9: Network lifetime with increasing node density.

Our major interest for energy-conserving study is to extend network lifetime by conserving resources. BECA demonstrates that we can avoid needlessly keeping the radio on longer by using information from above the MAC level. AFECA takes conservation a step further: by observing the size of their neighborhood, nodes can avoid needlessly duplicating routing offered by equivalent adjacent nodes. The result is that as node density rises (for example, many people attend a meeting, or sensors are randomly deployed in an area of interest), the network lasts longer, rather than unnecessarily exhausting itself through duplicated work. A corollary is that, with AFECA, one can simply "throw down" additional nodes to improve network lifetime.

5 Related Work

Our work builds on related work for radio energy models ad hoc routing protocols, and energy-aware MAC and application-level protocols.

Energy models: Stemm and Katz have analyzed power drain of WaveLAN network interfaces. In their model the cost of packet reception is only slightly more costly than listening to an idle channel, while sending costs 1.4x idle. As we discussed in Section 2, the choice of an energy model including idle-time consumption is very important in choosing how to change algorithms to conserve energy. Our work therefore uses the Stemm and Katz model.

Ad hoc routing protocols: A number of routing protocols have been proposed to provide multi-hop communication in wireless, ad hoc networks [14, 4, 15, 13]. Traditionally these protocol are evaluated in terms of packet loss rates, routing message overhead, and route length [3, 11, 7]. We evaluate our protocols by these metrics for comparison, and we add measures of power consumption and network lifetime to consider power consumption as well.

Both Chang and Tassiulas [5] and Pottie et al. [16] have recently suggested that one might select routes in an ad hoc network based on available energy. The effect of this work would be longer network lifetime. Our approach is to conserve energy by powering radios off rather than managing a fixed energy consumption, so our work work work complements their effort.

Heinzelman et al. present a set of protocols for communication in sensor networks based on flooding [10]. They examine the energy consumption of these protocols and show that suppressing duplicate transmissions of the same data can save power as calculated from a simple energy model (not considering energy consumption while radios are idle). Unlike their work, we consider more accurate power models and ad hoc routing protocols rather than flooding. These differences result in much different optimizations. We also consider optimizations based on adaptive fidelity that are specific to dense networks.

Energy-aware MAC protocols: PAMAS is a MAC-level protocol where radios power off when not actively transmitting or receiving packets [19, 18, 20]. PAMAS avoids the overhearing problem we discuss in Section 2, but it does not address the problem of energy consumption when nodes are idle. Solutions to overhearing are relevant, but for radios with high idle power consumption work such as we propose will be necessary.

TDMA protocols have been proposed to reduce energy consumption in sensor networks [16]. By reducing the duty cycle these protocols can trade idle-time energy consumption for latency. We believe TDMA MAC protocols will very important for power-constrained networks. Although we have not yet examined use of our approaches over TDMA protocols, our use of application-level information and node density can further improve power conservation.

IEEE 802.11 [12] supports ad hoc network configuration: mobile nodes are brought together to form a network on the fly. IEEE 802.11 also provides power management controls to allow disabling the transceiver to conserve energy. Although they specify how to turn off the radio, they do not discuss specific policies. We propose these policies assuming the presence of 802.11-like controls for basic and adaptive cases.

PicoNet: PicoNet proposes an integrated design of radios, small, battery powered nodes, and MAC and application protocols that minimize power consumption [2]. They reduce power consumption with a very low, application-dependent duty cycle (their paper does not specify, but presentations suggest intermittent polling with periods of 50 to 100s of seconds). They primarily use local base stations instead of multi-hop wireless routing, and assume frequent or continuous node movement. Their approaches are promising, but we are not aware of a detailed study if PicoNet power consumption. Our work differs from theirs by building on existing ad hoc routing protocols and by making use of adaptive fidelity to reduce power in dense node configurations.

6 Future Work

We have identified a number of areas for future work.

Most important is an understanding of these algorithms under different conditions: different traffic loads or kinds of traffic, higher levels of node mobility, coordinated (rather than random) node movement, and different levels of node density.

We have only consider one example of an adaptive fidelity algorithm, with a single approach at estimating neighborhood size and adjusting sleep times. A fuller understand of both aspects of AFECA are needed. We also plan to explore the concept of adaptive fidelity in other contexts. For example, we would like to understand the performance difference of sensing in dense sensor networks when sensors are only enabled with some duty cycle.

Additional exploration of network behavior as nodes fail is important. When does the network partition? How do mixes of nodes with different power affect the results? These results are likely sensitive to traffic mix.

Finally, experimentation is needed to validate these results with physical hardware in actual scenarios.

7 Conclusions

We have demonstrated two approaches to energy conservation for ad hoc routing. Power consumption in current wireless networks is idle-time dominated, so both focus on turning the radio off as much as possible.

BECA, our basic algorithm, uses routing- and application-layer information to achieve up to a 50% duty cycle. Although route setup latency increases, for sleep times of 10s we see energy savings of 40%.

Our second algorithm, AFECA, demonstrates *adaptive fidelity*. It adapts sleep times based on node density, scaling back node duty cycles (and so reducing routing "fidelity") when many interchangeable nodes are present. We have shown that it performs at least as well as BECA for packet loss, route latency, and energy in typical conditions, and it can nearly double network lifetime as density rises.

The common thread to these approaches is avoiding unnecessary energy consumption. In BECA we turn the radio off because it's unneeded and we'll check again later; with AFECA, we turn an unneeded radio off because our neighbors can check for us. These algorithms will be important to maximize the utility of networks of battery-powered embedded devices. The simple "add more to improve service" behavior of AFECA and adaptive fidelity are particularly important as the numbers of embedded devices and the ratio of devices to humans increases.

References

[1] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala. Improving simulation for network research. Technical Report 99-702b, USC, March 1999. Revised September 1999, to appear in IEEE Computer.

[2] Frazer Bennett, David Clarke, Joseph B. Evans, Andy Hopper, Alan Jones, and David Leask. Piconet: Embedded mobile networking. *IEEE Personal Communications Magazine*, 4(5):8–15, October 1997.

[3] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, October, 1998.

[4] Josh Broch, David B. Johnson, and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. *INTERNET-DRAFT, draft-ietf-manet-dsr-03.txt,*, October 1999. Work in progress.

[5] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networking. In *Proceedings of the IEEE Infocom*, page to appear, Tel Aviv, Israel, March 2000. ACM/IEEE.

[6] M.S. Corson and V. Park. An internet MANET encapsulation protocol (IMEP). *INTERNET-DRAFT, draft-ietf-manet-spec-01.txt*, August 1998. Work in progress.

[7] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Infocom*, March 2000.

[8] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, USA, August 1999. ACM.

[9] VINT group. UCB/LBNL/VINT network simulator—ns (version 2). `http://mash.cs.berkeley.edu/ns`.

[10] Wendi R. Heinzelman, Hoanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 175–185, August, 1999.

[11] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 195 – 206, August, 1999.

[12] LAN MAN Standards Committee of the IEEE Computer Society. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, 1999. IEEE Std 802.11, 1999 Edition.

[13] V. Park and S. Corson. Temporally-ordered routing algorithm (tora) version 1 functional specification. *INTERNET-DRAFT, draft-ietf-manet-tora-spec-02.txt,*, October 1999. Work in progress.

[14] Charles Perkins. Ad hoc on demand distance vector(aodv) routing. *Internet-Draft, draft-ietf-manet-aodv-04.txt*, October 1999,Work in progrss.

[15] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM SIGCOMM*, pages 234–244, Auguest 1994. A revised version of the paper is available at `http://www.cs.umd.edu/projects/mcml/papers/Sigcomm93.ps`.

[16] Gregory J. Pottie and William J. Kaiser. Wireless integrated network sensors (WINS): Principles and practice. To appear, CACM, 2000.

[17] The CMU Monarch Project. The cmu monarch projects wireless and mobility extension to ns. `http://www.monarch.cs.edu`.

[18] S. Singh and C.S. Raghavendra. PAMAS: Power aware multi-access protocol with signalling for ad hoc networks. 28(3):xxx, July 1998.

[19] S. Singh and C.S. Raghavendra. Pamas - power aware multi-access protocol with signalling for ad hoc networks. *ACM Computer Communication Review*, July 1998.

[20] S. Singh, M. Woo, and C.S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190, October, 1998.

[21] Mark Stemm and Randy H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. E80-B(8):1125–1131, August 1997.