Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks

Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts, 02139
{benjie, jamieson, hari, rtm}@lcs.mit.edu

Abstract

This paper presents *Span*, a power saving technique for multi-hop ad hoc wireless networks that reduces energy consumption without significantly diminishing the capacity or connectivity of the network. Span builds on the observation that when a region of a shared-channel wireless network has a sufficient density of nodes, only a small number of them need be on at any time to forward traffic for active connections.

Span is a distributed, randomized algorithm where nodes make local decisions on whether to sleep, or to join a *forwarding backbone* as a *coordinator*. Each node bases its decision on an estimate of how many of its neighbors will benefit from it being awake, and the amount of energy available to it. We give a randomized algorithm where coordinators rotate with time, demonstrating how localized node decisions lead to a connected, capacity-preserving global topology.

Improvement in system lifetime due to Span increases as the ratio of idle-to-sleep energy consumption increases, and increases as the density of the network increases. For example, our simulations show that with a practical energy model, system lifetime of an 802.11 network in power saving mode with Span is a factor of two better than without. Span integrates nicely with 802.11—when run in conjunction with the 802.11 power saving mode, Span improves communication latency, capacity, and system lifetime.

1. Introduction

Minimizing energy consumption is an important challenge in mobile networking. Significant progress has been made on low-power hardware design for mobile devices that the wireless network interface is often a device's single largest consumer of power. Since the network interface may often be idle, this power could be saved by turning the radio off when not in use. In practice, however, this approach is not straightforward: a node must arrange to turn its radio on not just to receive packets addressed to it, but also to participate in any higher-level routing and control protocols. The requirement of cooperation between power saving and routing protocols is particularly acute in the case of multi-hop ad hoc wireless networks, where nodes must forward packets for each other. Coordination of power saving with routing in ad hoc wireless networks is the subject of this paper.

This research was funded in part by NTT Corporation under the NTT-MIT collaboration and by Intel Corporation.

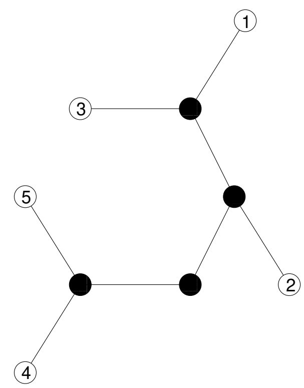

Figure 1: A connected backbone does not necessarily preserve capacity. In this connected topology, black nodes are coordinators. Packets between nodes 3 and 4 may contend for bandwidth with packets between nodes 1 and 2. On the other hand, if node 5 were a coordinator, no contention would occur.

A good power-saving coordination technique for wireless ad-hoc networks ought to have the following characteristics. It should allow as many nodes as possible to turn their radio receivers off most of the time, since even an idle receive circuit can consume almost as much energy as an active transmitter. On the other hand, it should forward packets between any source and destination with minimally more delay than if all nodes were awake. This implies that enough nodes must stay awake to form a connected backbone. Furthermore, the backbone formed by the awake nodes should provide about as much total capacity as the original network, since otherwise congestion may increase. This means that paths that could operate without interference in the original network should be represented in the backbone. For example, Figure 1 illustrates a topology that violates this principle. A good coordination technique should also not make many assumptions about the link layer's facilities for sleeping; it should work with any link-layer that provides for sleeping and periodic polling, including 802.11's ad-hoc power saving mode. Finally, power saving should inter-operate correctly with whatever routing system the ad-hoc network uses.

The algorithm presented in this paper, *Span,* fulfills the above requirements. Each node in the network running Span makes periodic, local decisions on whether to sleep or stay awake as a *coordinator* and participate in the forwarding backbone topology. To preserve capacity, a node decides to volunteer to be a coordinator if it discovers that two of its neighbors cannot communicate with each other directly or through an existing coordinator. To keep the number of redundant coordinators low and rotate this role amongst all nodes, each node delays announcing its willingness with a random delay that takes two factors into account: the amount of remaining battery energy, and the number of pairs of neighbors it can connect together. This combination ensures, with high probability, a capacity-preserving connected backbone at any point in time, where nodes tend to consume energy at about the same rate. Span does all this using only local information, consequently scaling well with the number of nodes. Our simulation results show that system lifetime with Span is more than a factor of two better than without Span, for a range of node densities, without much reduction in overall forwarding capacity.

The rest of the paper describes and evaluates Span. Section 2 reviews related work. Section 3 describes Span's algorithms and its interactions with the link layer. Section 4 presents our implementation of Span on top of an IEEE 802.11 link layer in the ns-2 network simulator. Section 5 presents performance results of several experiments. Finally, Section 6 concludes.

2. Related Work

The recent GAF [28] scheme of Xu *et al.* has similar goals to those of Span. In GAF, nodes use geographic location information to divide the world into fixed square grids. The size of each grid stays constant, regardless of node density. Nodes within a grid switch between sleeping and listening, with the guarantee that one node in each grid stays up to route packets. Span differs from GAF in two areas. First, unlike GAF, Span does not require that nodes know their geographic positions. Instead, Span uses broadcast messages to discover and react to changes in the network topology. Second, Span integrates with 802.11 power saving mode nicely: non-coordinator nodes can still receive packets when operating in power saving mode.

In AFECA [27], each node maintains a count of the number of nodes within radio range, obtained by listening to transmissions on the channel. A node switches between sleeping and listening, with randomized sleep times proportional to the number of nearby nodes. The net effect is that the number of listening nodes is roughly constant, regardless of node density; as the density increases, more energy can be saved. AFECA's constants are chosen so that there is a high probability that the listening nodes form a connected graph, so that ad hoc forwarding works. An AFECA node does not know whether it is required to listen in order to maintain connectivity, so to be conservative AFECA tends to make nodes listen even when they could be asleep. Span differs from AFECA in that with high likelihood, Span never keeps a node awake unless it is absolutely essential for connecting two of its neighbors. Furthermore, Span explicitly attempts to preserve the same overall system capacity as the underlying network where all nodes are awake, which ensures that no increase in congestion occurs.

The PAMAS power-saving medium access protocol [19, 24] turns off a node's radio when it is overhearing a packet not addressed to it. This approach is suitable for radios in which processing a received packet is expensive compared to listening to an idle radio medium. Kravets and Krishnan [13] present a system in which mobile units wake up periodically and poll a base station for newly arrived packets. Like Stemm and Katz [25], they show that setting the on/off periods based on application hints reduces both power and delay. Span assumes the presence of an ad hoc polling mechanism such as that provided by 802.11, and could potentially work in concert with application hints; such hints would apply only to sleeping nodes, not coordinators. Smith *et al.* [15] propose an ad hoc network that elects a virtual base station to buffer packets for local nodes. They do not, however, attempt to make sure that enough of these base stations are present to preserve connectivity in a multi-hop ad hoc network.

Minimum-energy routing [23] saves power by choosing paths through a multi-hop ad hoc network that minimize the total transmit energy. This approach has been extended by Chang and Tassiulas [4] to maximize overall network lifetime by distributing energy consumption fairly. In this protocol, nodes adjust their transmission power levels and select routes to optimize performance. Ramanathan and Rosales-Hain describe distributed algorithms that vary transmission power and attempt to maintain connectedness [20]. Rodoplu and Meng give a distributed algorithm to produce minimum-power routes by varying node transmission power [21]. Wattenhofer *et al.* [26] describe a topology maintenance algorithm using similar underlying radio support, but their algorithm guarantees global connectedness using directional information. Span controls whether or not the receiver is powered on, rather than controlling the transmit power level. It also pays close attention to overall system capacity, in addition to maintaining connectivity.

An alternative approach is described by Heinzelman *et al.*, whose LEACH protocol selects rotating cluster-heads to collect local information and transmit it to a base station in a wireless sensor network [10]. Like LEACH, Intanagonwiwat *et al.*'s directed diffusion mechanism [11] takes advantage of aspects of sensor networks, particularly the possibility of aggregating and compressing data, that are not present in general-purpose networks.

In general, the basic idea that a path with many short hops is sometimes more energy-efficient than one with a few long

Routing Layer	**GPSR**	**DSR**	**AODV**
	Span		
MAC/PHY	**802.11**		

Figure 2: Span is a protocol that operates under the routing layer and above the MAC and physical layers. The routing layer uses information Span provides, and Span leverages any power saving features of the underlying MAC layer.

hops could be applied to any ad hoc network with variable-power radios and knowledge of positions. This technique and Span's are orthogonal, so their benefits could potentially be combined.

3. Span Design

Span adaptively elects "coordinators" from all nodes in the network. Span coordinators stay awake continuously and perform multi-hop packet routing within the ad hoc network, while other nodes remain in power-saving mode and periodically check if they should wake up and become a coordinator.

Span achieves four goals. First, it ensures that enough coordinators are elected so that every node is in radio range of at least one coordinator. Second, it rotates the coordinators in order to ensure that all nodes share the task of providing global connectivity roughly equally. Third, it attempts to minimize the number of nodes elected as coordinators, thereby increasing network lifetime, but without suffering a significant loss of capacity or an increase in latency. Fourth, it elects coordinators using only local information in a decentralized manner—each node only consults state stored in local routing tables during the election process.

Span is proactive: each node periodically broadcasts `HELLO` messages that contain the node's status (i.e., whether or not the node is a coordinator), its current coordinators, and its current neighbors. These `HELLO` messages, and consequently the states maintained by Span, are similar to those of proactive ad hoc routing protocols (e.g., geographic routing [8, 14, 12] or DSDV [18]). Each node maintains only a small amount of additional state—its coordinators and coordinators of neighbors—in addition to a list of neighbors normally found in the routing table.

As shown in Figure 3, Span runs above the link and MAC layers and interacts with the routing protocol. This structuring allows Span to take advantage of power-saving features of the link layer protocol, while still being able to affect the routing process. For example, non-coordinator nodes can periodically turn on their radios and listen (e.g. 802.11 power-saving mode [1]) or poll (as in LPMAC [15]) for their packets. Span leverages a feature of modern power-saving MAC layers, where if a node has been asleep for a while, packets destined for it are not lost but are buffered at an upstream neighbor. When the node awakens, it can retrieve these packets from the buffering node, typically a coordinator. Span also requires a modification to the route lookup process at each node—at any time, only those entries in a node's routing table that correspond to currently active coordinators can be used as valid next-hops (unless the next hop is the destination itself).

A Span node switches state from time to time between being a coordinator and being a non-coordinator. A node includes its current state in its `HELLO` messages. The following sections describe how a node decides that it should announce that it is a coordinator, and how it decides that it should withdraw from being a coordinator.

3.1 Coordinator Announcement

Periodically, a non-coordinator node determines if it should become a coordinator or not. The following *coordinator eligibility rule* in Span ensures that the entire network is covered with enough coordinators:

Coordinator eligibility rule: if two neighbors of a non-coordinator node cannot reach each other either directly or via one or two coordinators, the node should become a coordinator.

While this election algorithm does not yield the minimum number of coordinators required to merely maintain connectedness, it forms a network that roughly contains a coordinator in every populated radio range in the entire network topology. Since packets will be routed through coordinators, this topology ought to yield good capacity.

Announcement contention occurs where multiple nodes discover the lack of a coordinator at the same time, and all decide to become a coordinator. We resolve contention by delaying coordinator announcements with a randomized back-off delay. Each node chooses a delay value, and delays the `HELLO` message that announces the node's volunteering as a coordinator for that amount of time. If at the end of the delay, the node has not received any `HELLO` messages from other potential coordinators, it sends the `HELLO` message. Otherwise, it reevaluates its eligibility based on any `HELLO` messages received, and makes its announcement if and only if the eligibility rule still holds.

We consider a variety of factors in our derivation of the back-off delay. Consider first the case when all nodes have roughly equal energy, which implies that only topology should play a role in deciding which nodes become coordinators. Let N_i be the number of neighbors for node i and let C_i be the number of additional pairs of nodes among these neighbors that would be connected if i were to become a coordinator and forward packets. Clearly, $0 \leq C_i \leq \binom{N_i}{2}$. We call $\frac{C_i}{\binom{N_i}{2}}$ the *utility* of node i. If nodes with high C_i become coordinators, fewer coordinators in total may be needed in order to make sure every node can talk to a coordinator; thus a node with a high C_i should volunteer more quickly than one with smaller C_i.

Now, if there are multiple nodes within radio range that all have the same utility, we need to ensure that not too many of them all become coordinators. This is because such coordinators are redundant—they do not increase system capacity, but simply drain power. If the potential coordinators make their decisions simultaneously, they may all decide to become coordinators. If, on the other hand, they decide one at a time, only the first few will become coordinators, and the rest will notice that there are already enough coordinators and go back to sleep. To handle this, we use a randomized "slotting-and-damping" method reminiscent of various multicast protocols, including XTP [5], IGMP [7] and SRM [9]: choose the delay for each node randomly over an interval proportional to $N_i \times T$, where T is the round-trip delay for a small packet over the wireless link.

Thus, when all nodes have roughly equal energy, the above discussion suggests a backoff delay of the form:

$$delay = \left(\left(1 - \frac{C_i}{\binom{N_i}{2}} \right) + R \right) \times N_i \times T \qquad (1)$$

The randomization is achieved by picking R uniformly at random from the interval $[0, 1]$.

Now consider the case when nodes may have unequal energy left in their batteries. We observe that what matters in a heterogeneous network is not necessarily the absolute amount of energy available at the node, but the amount of energy scaled to the maximum amount of energy that the node can have. Let E_r denote the amount of energy (in Joules) at a node that still remains, and E_m be the maximum amount of energy available at the same node. A reasonable (but not the only) notion of fairness can be achieved by ensuring that a node with a larger value of E_r/E_m is more likely to volunteer to become a coordinator more quickly than one with a smaller ratio. Thus, we need to add a decreasing function of E_r/E_m that reflects this, to Equation 1. There are an infinite number of such functions, from which we choose a simple linear one: $1 - E_r/E_m$. In addition to its simplicity, this choice is attractive because it ensures that the rate with which a node reduces its propensity to advertise (as a function of the amount of energy it has left), is constant. (We experimented with a few other functions, including an exponentially decaying function of E_r/E_m and an inversely decaying function of E_r/E_m; the simple linear one worked best.)

Combining this with Equation 1 yields the following equation for the backoff delay in Span:

$$delay = \left(\left(1 - \frac{E_r}{E_m} \right) + \left(1 - \frac{C_i}{\binom{N_i}{2}} \right) + R \right) \times N_i \times T \quad (2)$$

Observe that the first term does not have a random component; thus if a node is running low on energy, its propensity to become a volunteer is guaranteed to diminish relative to other nodes in the neighborhood with similar neighbors.

In a network with uniform density and energy, our election algorithm rotates coordinators among all nodes of the network. It achieves fairness because the likelihood of becoming a coordinator falls as a coordinator uses up its battery. In practice, however, ad hoc networks are rarely uniform. Our announcement rule adapts to non-uniform topology: a node that connects network partitions together will always be elected a coordinator. This property preserves capacity over the lifetime of the network. Because of Span's emphasis on capacity-preservation to the extent possible, such critical nodes will unavoidably die before other less-critical ones. However, in a mobile Span network, a given node is rarely stuck in such a position, and this improves fairness dramatically.

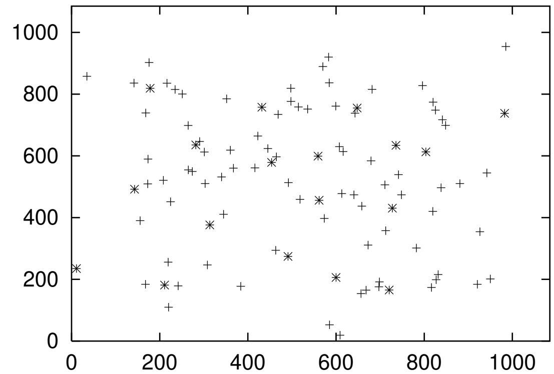

Figure 3: A scenario with 100 nodes, 18 coordinators, and a radio range of 250 meters. The nodes marked "*" are coordinators; the nodes marked "+" are non-coordinator nodes.

3.2 Coordinator Withdrawal

Each coordinator periodically checks if it should withdraw as a coordinator. A node should withdraw if every pair of its neighbors can reach each other either directly or via some other coordinators. However, in order to also ensure fairness, after a node has been a coordinator for some period of time, it withdraws if every pair of neighbor nodes can reach each other via some other neighbors, even if those neighbors are not currently coordinators. This rule gives other neighbors a chance to become coordinators.

To prevent temporary loss of connectivity between a coordinator's withdrawal message and the announcement from a new coordinator, a node continues to serve as a coordinator for a short period of time even after announcing its withdrawal. This "grace period" allows the routing protocol to continue to use the old coordinator until a new coordinator is elected.

Figure 3 shows the result of our election algorithm at a random point in time on a network of 100 nodes in a 1000 meter × 1000 meter area, where each radio has an isotropic circular range with a 250 meter radius.

4. Simulator Implementation

This section describes our implementation of Span, geographic forwarding, the 802.11 power saving mode (with our own improvements), and the energy model we used in our simulations. We ran our Span implementation in the `ns-2` network simulator environment.

4.1 Span and Geographic Forwarding

Our implementation uses the geographic forwarding algorithm. We chose to implement geographic forwarding primarily because of its simplicity; Span can be used with other routing protocols as well.

Span's election algorithm requires each node to advertise its coordinators, its neighbors, and whether it is a coordinator or not. To reduce protocol overhead, we piggyback Span `HELLO` information (see Section 3.1) onto the broadcast updates required by geographic forwarding.

Geographic forwarding is greedy. The source node annotates each packet with the geographic location of the destination node. Upon receiving a packet for a node not in radio range, a coordinator forwards the packet to a neighboring coordinator that is closest to the destination. If no such coordinator exists, the packet is forwarded to a non-coordinator that is closer to the destination. Otherwise, we know that a packet has encountered a void, and so it is dropped. (We did not implement GPSR [12], which ameliorates the effects of voids.)

Our simulations do not use a location service. Instead, each sender uses the `GOD` module of `ns` to obtain the geographic location of the destination node. Hence, our simulation results may be better than one might expect with a real location service, such as GLS [14].

Our geographic forwarding algorithm also implements MAC-layer failure feedback and interface queue traversal [2, 12]. These mechanisms allow the routing layer to readily remove unresponsive nodes from its routing table and rescue packets using these nodes as the next hop.

4.2 Coordinator Election

Aside from the coordinator election algorithm discussed in Section 3, we implemented two special cases for electing coordinators.

- Because the source and destination nodes of each flow need to constantly send and receive packets, they do not operate in power saving mode. Thus, they automatically become coordinators.

- The geographic routing algorithm can readily detect that a coordinator has left the region through MAC layer failure feedback. However, the Span election algorithm may not react fast enough to elect new coordinators. In the worst case, nodes must wait until the old coordinator information has expired before a new coordinator can be elected. Because geographic forwarding falls back to using non-coordinators to route packets if coordinators do not exist, a non-coordinator node announces itself as a coordinator if it has received a large number of packets to route in the recent past. If this coordinator turns out to be redundant, the coordinator withdraw algorithm will force the node to withdraw itself as a coordinator soon after.

4.3 802.11 Ad Hoc Power-saving Mode

Span determines when to turn a node's radio on or off, but depends on the low level MAC layer to support power saving functions, such as buffering packets for sleeping nodes. We have implemented Span on top of the 802.11 MAC and physical layers with ad hoc power saving support [1].

802.11 ad hoc power-saving mode uses periodic *beacons* to synchronize nodes in the network. Beacon packets contain timestamps that synchronize nodes' clocks. A beacon period starts with an ad hoc traffic indication message window (*ATIM window*), during which all nodes are listening, and pending traffic transmissions are advertised. A node that receives and acknowledges an advertisement for unicast or broadcast traffic directed to itself must stay on for the rest of the beacon period. Otherwise, it can turn itself off at the end of the ATIM window, until the beginning of the next beacon period. After the ATIM window, advertised traffic is transmitted. Since traffic cannot be transmitted during the ATIM window, the available channel capacity is reduced.

When the 802.11 MAC layer is asked to send a packet, it may or may not be able to send it immediately, depending on which ATIMs have been sent and acknowledged in the immediately preceding or current, ATIM window. If the packet arrives at the MAC during the ATIM window, or if the advertisement for the packet has not been acknowledged, it needs to be buffered. In our implementation, we buffer packets for two beacon periods. Packets that have not been transmitted after two beacon periods are dropped.

The beacon period and ATIM window size affect routing performance greatly [22]. While using a small ATIM window may improve energy savings, there may not be enough time for all buffered packets to be advertised. Using an ATIM window that is too large not only decreases available channel utilization, it may also not leave enough room between the end of the ATIM window and the beginning of the next beacon period to transmit all advertised traffic. We have experimentally determined that a beacon period of 200 ms and an ATIM window size of 40 ms result in good throughput and low loss rate.

Aside from decreased channel capacity, 802.11 power saving mode (without Span) also suffers from long packet delivery latency: for each hop that a packet traverses, the packet is expected to be delayed for half a beacon period.

4.4 Improving 802.11 using Span

Using Span on top of 802.11 ad hoc power saving mode can improve routing throughput and packet delivery latency. Because coordinators do not operate in power saving mode, packets routed between coordinators do not need to be advertised or delayed. To further take advantage of the synergy between Span and 802.11 power saving mode, we have made the following modifications to our simulation of 802.11 power saving mode.

- **No advertisements for packets between coordinators.** Packets routed between coordinators are marked by Span. While the MAC layer still needs to buffer these packets if they arrive during the ATIM window, it does not send traffic advertisements for them. To ensure that Span does not provide incorrect information due to topology changes, the MAC maintains a separate neighbor table. The MAC layer uses a bit in the MAC header of each packet it sends to notify neighbors of its power saving status. Since the MAC layer can sniff the header of every packet, including RTS packets, this neighbor table is likely to be correct. When a node withdraws as a coordinator, advertisements for traffic to that node will be sent during the next ATIM window. This optimization allows the ATIM window to be reduced without hurting throughput.

- **Individually advertise each broadcast message.** With unmodified 802.11 power saving mode, a node only needs to send one broadcast advertisement even if it has more than one broadcast message to send. This is because once a node hears an advertisement for a broadcast message, it stays up for the entire duration of the beacon period. Since most traffic to non-coordinator nodes in our network would be broadcast messages sent by Span and the geographic routing protocol, we modified the MAC so each broadcast message must be explicitly advertised. For example, if a node receives 5 broadcast advertisements, no unicast advertisements, and then 5 broadcast messages after the ATIM window, it can safely turn itself off.

- **New advertised traffic window.** With unmodified 802.11 power saving mode, if a node receives a unicast advertisement, it must remain on for the rest of the beacon period. In a Span network, packets routed via non-coordinator nodes are rare. To take advantage of this, we introduced a new *advertised traffic window* in the MAC. The advertised traffic window is smaller than the beacon period. It starts at the beginning of the beacon period, and extends beyond the end of the ATIM window. Outside the ATIM window but inside the advertised traffic window, advertised packets and packets to coordinators can be transmitted. Outside the advertised traffic window, however, only packets between coordinators can be transmitted. This allows a node in power saving mode to turn itself off at the end of the advertised traffic window until the next beacon period.

These three modifications allow each node to use a long beacon period and a short ATIM window. The short ATIM window improves channel utilization, while the long beacon period increases the fraction of time a non-coordinator node can remain asleep. In our simulations, we used a beacon period of 300 ms, an ATIM window of 20 ms, and an advertised traffic window of 100 ms. We set the propagation delay T in Equation 2 to be the length of a beacon period.

Section 5 compares performance of Span with the modified 802.11 power saving mode, unmodified 802.11 in ad hoc power saving mode, and unmodified 802.11 without power saving mode.

Tx	Rx	Idle	Sleeping
1400 mW	1000 mW	830mW	130mW

Table 1: Power consumption of the Cabletron 802.11 network card in the "Tx" (transmit), "Rx" (receive), "Idle," and "Sleeping" modes.

4.5 Energy Model

To accurately model energy consumption, we took measurements of the Cabletron Roamabout 802.11 DS High Rate network interface card (NIC) operating at 2 Mbps in base station mode. To measure power consumed by the card, we powered a portable computer solely with its AC adapter (without the battery), and measured the voltage across a resistor placed in series with the card on the computer to obtain the instantaneous current through the NIC. The voltage across the NIC remained constant at all times, thus from the instantaneous current measurement, we calculated the instantaneous power consumed by the card. We summarize the time-averaged results in Table 1, and note that these closely match the results obtained by Feeney and Nilsson [6] for similar 802.11 network interface cards in the ad hoc mode.

We obtained the "Rx" state measurement by putting the card into non-power saving mode, and measuring the power required to listen for a packet, decode it, and pass its contents up to the host. The "idle" state measurement was obtained in the same manner, but measuring only the power required to listen for a packet. In contrast, the "sleep" state measurement was obtained by putting the card into power saving mode, and measuring the average (lower, and near-constant) power consumption during the part of the power saving cycle where the card was not listening for packets. The key point to note is the large difference between the power consumption of idle and sleeping modes.

5. Performance Evaluation

To measure the effectiveness of Span, we simulated Span, with geographic forwarding, on several static and mobile topologies. Simulation results show that Span not only performs well by extending network lifetime, it out-performs unmodified 802.11 power saving network in handling heavy load, per-packet delivery latency, and network lifetime.

5.1 Simulation Environment

We simulated Span in the `ns-2` [17] network simulator using the CMU wireless extensions [16]. The geographic forwarding algorithm, as described in Section 4.1, routes packets from source to destination. Span runs on top of 802.11 MAC layer with power saving support and modifications described in Section 4.3. In this section, we compare performance of Span against both unmodified 802.11 MAC in power saving mode and unmodified 802.11 MAC not in power saving mode. For convenience, we will refer to them as Span, 802.11 PSM, and 802.11.

To evaluate Span in different node densities, we simulate 120-node networks in square regions of different sizes. Nodes in our simulations use radios with a 2 Mbps bandwidth and 250 meters nominal radio range. Twenty nodes send and receive traffic. Each of these nodes send a CBR flow to another node, and each CBR flow sends 128 byte packets. In Section 5.2 we vary the rate of the CBR traffic to measure performance of Span under different traffic load. In other experiments, each sender sends three packets per second, for a total of 60 Kbps of traffic.

To ensure that the packets of each CBR flow go through multiple hops before reaching the destination node, 10 source and destination nodes are placed, uniformly at random, on each of two 50 meter-wide, full-height strips located at the left and right of the simulated region. A source must send packets to a destination node on the other strip. The initial positions of the remaining 100 nodes are chosen uniformly at random in the entire simulated region. Thus, the square root of the area of the simulated region and the number of hops needed by each packet are approximately proportional.

Source and destination nodes never move. In mobile experiments, the motion of the remaining 100 nodes follows the *random waypoint* model [3]: initially, each node chooses a destination uniformly at random in the simulated region, chooses a speed uniformly at random between 0 and 20 m/s, and moves there with the chosen speed. The node then pauses for an adjustable period of time before repeating the same process. The degree of mobility is reflected in the pause time. By default, we used a pause time of 60 seconds.

For simplicity, we did not use a location service in our simulations. Instead, a router obtains the location of the destination node from the `GOD` module in `ns`. Since the location lookup is only required once per flow at the sender, we believe the overhead produced by the location service is not likely to change our results. Nevertheless, location services such as GLS [14] can be used with Span.

All experimental results in this section are averages of five runs on different randomly-chosen scenarios. We define *node density* (as used in our graph axis labels) as the number of nodes that are not sources or destinations per radio range, an area of $250^2 \times \pi$ square meters.

5.2 Capacity Preservation

One of Span's goals is to preserve total network capacity, by making sure that if there are non-conflicting paths in the underlying network, there are similar non-conflicting paths in the coordinator backbone. This section compares the capacity available in a Span network with the capacity in an ordinary 802.11 network. We measure capacity by the number of packets the network can successfully deliver per unit time; capacity is inversely proportional to the network's packet loss rate. Additionally, we show that despite using fewer nodes to forward packets, Span does not significantly increase delivery latency and number of hops each packet traverses.

Figure 4 shows packet delivery rate as the bit rate of each CBR flow increases. There is no motion in these simulations. The simulation region has an area of 1000 meters × 1000 meters. On average, each packet traverses 6 hops.

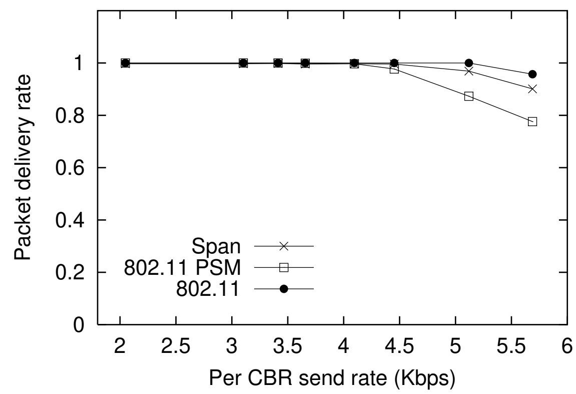

Figure 4: Packet delivery rate as a function of per-CBR-flow bit rate. Each packet traverses six hops. Under higher traffic load, Span delivers more packets than 802.11 PSM, but slightly less than 802.11.

Unmodified 802.11 PSM drops significantly more packets when the CBR flow rate increases past 4 Kbps. Most of these packet drops occur either because the ATIM window is not long enough to allow all buffered unicast packets to be advertised, or because after the ATIM window there is not enough time until the start of the next beacon period for all advertised packets to be transmitted. After two beacon periods of buffering, all packets are dropped by the MAC. Because Span does not need to advertise traffic between coordinators and uses a shorter ATIM window and longer beacon period, Span delivers more packets.

Span has higher loss rates than regular 802.11 when the bit rate increases beyond 4.5 Kbps. This increase in loss rate is largely due to the fact that Span uses a 20 ms ATIM window per 300 ms beacon period, which reduces utilization by 6.7%. Additionally, using fewer nodes to forward packets may decrease potential channel utilization even more: each time a node exponentially backs off to avoid collision, there is a greater chance that the channel becomes unoccupied for a longer period of time.

Table 2 shows the routing behavior and loss rates of Span, 802.11 PSM, and 802.11 with a 3 Kbps per CBR flow rate. We vary the simulation area to change node density and the number of hops each packet needs to traverse. There is no motion in these simulations. Despite using fewer nodes to forward packets, Span delivers packets using only a slightly higher number of hops. Span's packet delivery latency is higher than that of 802.11, but significantly lower than that of 802.11 PSM. With 802.11 PSM, each hop accounts for roughly 200 ms of latency, which corresponds with the 200 ms beacon period used.

Geographic forwarding using Span coordinators encounters fewer voids: coordinators are elected to connect neighboring nodes, and are therefore unlikely to occur at the edge of a void. Thus, Span has a lower loss rate than both 802.11 and 802.11 PSM when the node density is low.

		Span			802.11 PSM			802.11		
Area	Density	Loss	Lat (ms)	Hops	Loss	Lat (ms)	Hops	Loss	Lat (ms)	Hops
500m×500m	78.5	0.0%	23.4	2.8	0.0%	423	2.4	0.0%	5.69	2.4
750m×750m	34.9	0.0%	30.7	4.5	0.0%	739	4.0	0.0%	11.2	4.0
1000m×1000m	19.6	0.4%	40.5	6.1	0.1%	1032	5.4	0.0%	16.9	5.4
1250m×1250m	12.6	1.9%	45.2	7.8	10.7%	1391	7.3	7.0%	20.6	7.3

Table 2: Performance of geographic forwarding with Span, 802.11 PSM, and 802.11 as node density and area of simulation region changes. Span delivers packets using slightly more hops. Span's packet delivery latency is higher than 802.11's, but is significantly less than that of 802.11 PSM.

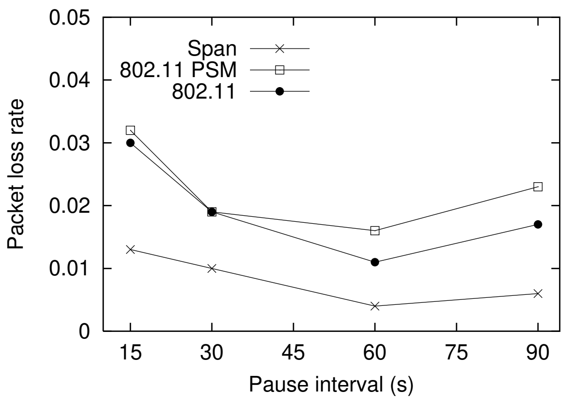

Figure 5: Packet loss rate as a function of pause time. The simulation area is a 1000 meter × 1000 meter square. Mobility does not affect Span very much, and geographic forwarding with Span delivers more packets than with 802.11 PSM and 802.11 because it encounters fewer voids.

These results show that Span does not significantly degrade network capacity, and can forward more packets than 802.11 PSM under high load. Furthermore, Span increases packet latency only slightly, despite using a fewer number of nodes to forward packets.

5.3 Effects of Mobility

Figure 5 shows the effects of mobility on packet loss rate. In these simulations an area of 1000 meters × 1000 meters is used. Each simulation lasts 400 seconds. Nodes follow the random waypoint motion model, and the length of the pause time reflects the degree of mobility.

The degree of mobility does not significantly affect routing with Span coordinators. Span consistently performs better than both 802.11 PSM and 802.11. Most packet drops in these simulations are caused by temporary voids created by mobility. Because geographic forwarding with Span encounters fewer voids, its loss rate is lower.

5.4 Coordinator Election

Ideally, Span would choose just enough coordinators to preserve connectivity and capacity, but no more; any coordinators above this minimum just waste power. This section compares the number of coordinators Span chooses with the

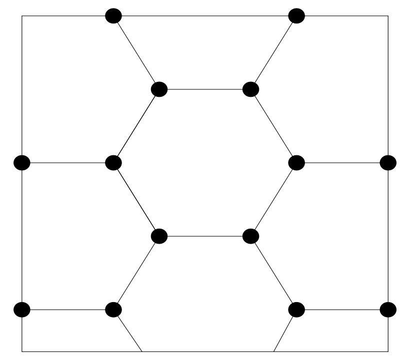

Figure 6: An approximation to an optimal layout of coordinators in a 1000 meter × 1000 meter area. There are 14 coordinators in this layout.

number that would be required to form a hexagonal grid layout, shown in Figure 6; the hex grid layout of nodes, while perhaps not optimal, produces a connected backbone in every direction with very few coordinators.

The hexagonal grid layout of coordinators place a coordinator at each vertex of a hexagon. Every coordinator can communicate with the three coordinators that it is connected to through an edge of a hexagon, which is 250 meters long (the radio range). Each hexagon has six coordinators, but each coordinator is shared by three hexagons. Therefore each hexagon is only responsible for two coordinators. Each hexagon has an area of 162,380 m^2. Thus, given a simulation area of d^2 meters, the number of coordinators expected in this area, C_{ideal} is

$$C_{ideal} = 2 \cdot \frac{d^2}{162380} \tag{3}$$

Figure 7 shows coordinator density as a function of node density. Coordinator density is computed from the average number of coordinators elected by Span over 300 seconds of five mobile simulations. Nodes that become coordinators only because they are sources or destinations are not counted. Because these nodes only reside at the left and right edges of the simulation area, numbers in the figure are only slightly less than the number of coordinators had there been no sources or sinks. Points on the "Ideal" curve in Figure 7 are computed using the ideal number of coordinators predicted by Equation 3.

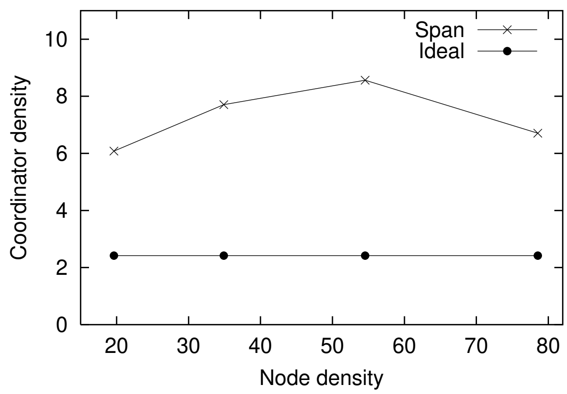

Figure 7: Ideal and actual coordinator density as a function of node density. Span elects more coordinators than the ideal case because of non-uniform density, coordinator rotation, and announcement collision.

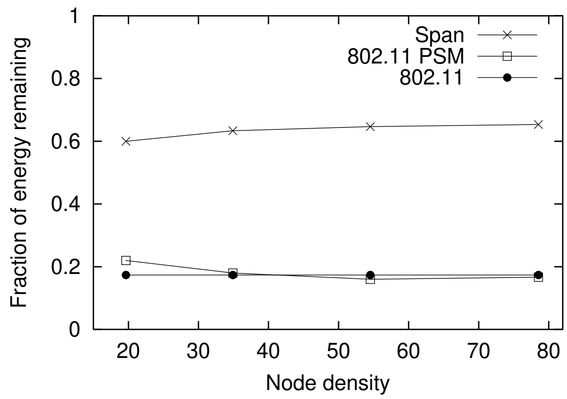

Figure 8: Fraction of energy remaining after 300 seconds of simulation. Span provides significant amount of savings over 802.11 PSM and 802.11.

Span elects more coordinators than are necessary. There are three reasons for this. First, non-uniform density often causes more nodes to become coordinators. Second, to rotate coordinators among all nodes, the optimal set of coordinators may not always be selected. Third, Equation 2 sometimes fails to prevent `HELLO` message collisions.

5.5 Energy Consumption

This section evaluates Span's ability to save energy. The potential for savings depends on node density, since the fraction of sleeping nodes depends on the number of nodes per radio coverage area. The energy savings also depends on a radio's power consumption in sleep mode and the amount of time that sleeping nodes must turn on their receivers to listen for 802.11 beacons and Span `HELLO` messages.

Figure 8 shows the fraction of energy remaining at each node after 300 seconds of simulation. Each node has an initial energy of 300 J. These trials are the same as those in Figure 7. From these results, we find that Span provides a considerable amount of energy savings over 802.11, while 802.11 PSM does not provide any energy savings at all. This is because geographic forwarding and Span need to send broadcast messages. With 802.11 PSM, each time a node receives a broadcast advertisement, it must stay up for the entire beacon period. This prevents non-coordinators from going back to sleep. When the node density is low, the number of broadcast messages in a radio range decreases, and 802.11 PSM yields a small amount of energy savings.

We also find that as density increases, a smaller fraction of the nodes are elected coordinators. Consequently, we expect energy savings to increase. In practice, however, energy savings do not increase as much. To understand why, we estimate the amount of energy used in a Span system based on an estimate of the average fraction of time a node must run its radio in idle mode. We call this fraction f_{idle}:

$$f_{idle} = \frac{C}{N} + \left(1 - \frac{C}{N}\right) \cdot f_{up} \tag{4}$$

In Equation 4, N is the total number of nodes, C is the number of coordinators elected, and f_{up} is the fraction of the time a node in sleep mode must wake up to listen for beacons and `HELLO` messages. Span uses a 20 ms ATIM window per 300 ms of beacon period. Thus the smallest value for f_{up} is 0.067. In the worst case, f_{up} can be as high as 0.333, when a non-coordinator node must stay up for the entire duration of the advertised traffic window (100 ms).

We define α as the ratio of the power consumption of the radio in sleep mode to the power consumption of the radio in idle mode. Then, using f_{idle}, the amount of energy savings can be estimated as

$$\frac{1}{f_{idle} + \alpha * (1 - f_{idle})} \tag{5}$$

Note that because f_{idle} depends on $\frac{C}{N}$, and that the coordinator density stays the same for different node densities, the gain in energy savings also depends on the node density.

Figure 9 plots Equation 5 as a function of α, substituting C_{ideal} and 0 as values for C and f_{up}. This figure shows that the amount of energy saving increases rapidly, as the value of α decreases. Our energy model uses $\alpha = 0.157$ from measurements. Figure 10 plots Equation 5 as a function of f_{up}, using C_{ideal} and 0.157 as values for C and α. This figure shows that as f_{up} increases, the gain in energy savings decreases as well. These two figures explain why in Figure 8, the gain in energy savings is a sub-linear function of node density.

We can calculate the actual values of f_{up} in our experiments using several statistics gathered from the first 300 seconds of the simulation. The numbers in the f_{up} column of Table 3 are calculated using Equation 4, using values from the "Idle time" column as f_{idle}. We substitute C/N with numbers in the "Time as coordinator" column divided by 300 seconds. The fact that f_{up} is very close to the maximum value of 0.333 at high density suggests that Span broadcast messages are expensive when density is high—the large number of broadcast messages per radio range keeps nodes awake for a longer period of time.

Density	Sleep time	Idle time	Time as coordinator	Energy remaining	Idle/Sleep energy	f_{up}
78.5	209 s	90 s	15.1 s	65.3%	66.3%	0.263
54.5	206 s	94 s	25.6 s	64.7%	65.1%	0.249
34.9	199 s	100 s	42.7 s	63.3%	63.7%	0.223
19.6	185 s	114 s	71.7 s	60.0%	60.4%	0.185

Table 3: Amount of time each node spends in sleep and idle mode, as a coordinator, and the energy remaining at each node as node density changes. The Idle/Sleep energy column shows the percent of energy remaining at each node calculated using only Sleep time and Idle time. It shows that the energy spent routing packets and sending broadcast messages are not significant. The f_{up} column shows the fraction of each beacon period that a node is awake. At higher densities, broadcast messages keep each node up for a longer period of time.

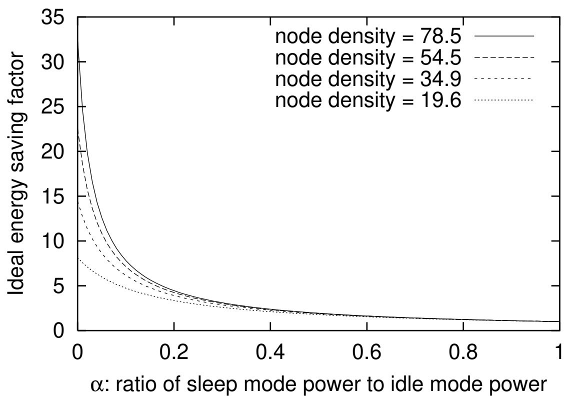

Figure 9: Energy savings as a function of α, computed using Equation 5, substituting C_{ideal} and 0 as values for C and f_{up} when computing f_{idle}. This figure shows that as α increases, the amount of energy saving decreases significantly. In our experiments, we used $\alpha = 0.16$.

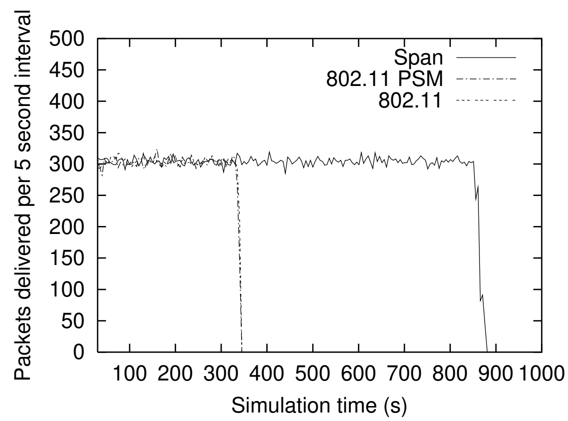

Figure 11: Packet delivery rate as a function of simulation time. This simulation ran on a 500m×500m area. With Span, the delivery rate does not drop below 90% until 851 seconds into the simulation, significantly longer than with 802.11 PSM and 802.11.

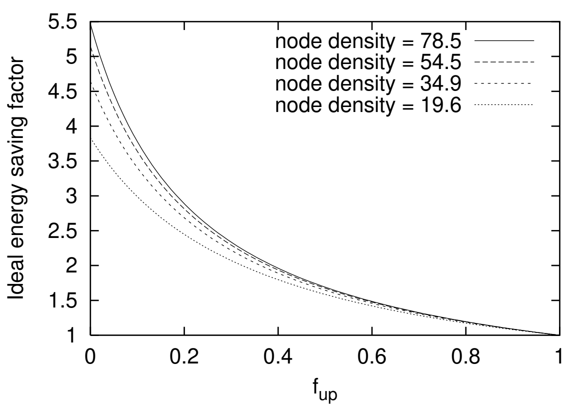

Figure 10: Energy savings as a function of f_{up}, computed using Equation 5, substituting C_{ideal} and 0.157 as values for C and α. This figure shows that as f_{up} increases, the amount of energy saving decreases as well. In our experiments, f_{up} was between 0.185 and 0.263.

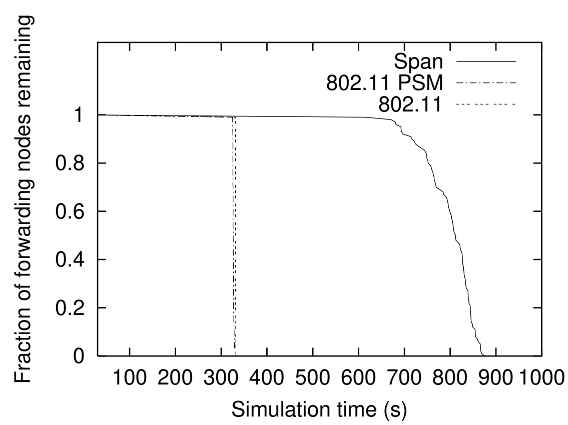

Figure 12: Fraction of nodes remaining as a function of simulation time. This figure is obtained from the same simulation as Figure 11. With Span, nodes remain alive for significantly longer periods of time.

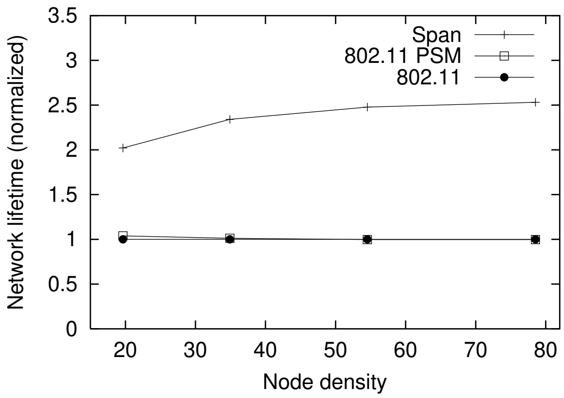

Figure 13: Network lifetime, defined as the time it takes for packet delivery rate to drop below 90%, as a function of node density. The figure is normalized to the lifetime of a 802.11 network. Network lifetime with Span is a factor of 2 better than without.

Finally, the numbers in the "Energy remaining" column in Table 3 correspond to the data points in Figure 8. The Idle/Sleep energy column shows the amount of energy remaining calculated using only the amount of time a node spends in sleep and idle mode. It does not taken into account of packet forwarding and sending and receiving broadcast messages. The fact that the numbers in the Idle/Sleep energy column are very close to those in the Energy remaining column suggest that routing packets does not consume much energy in comparison.

Results in this section show that Span saves energy by a factor of 3.5 or more over 802.11 PSM and 802.11. However, the amount of energy savings does not increase significantly as node density increases.

5.6 Network Lifetime

This section shows that Span distributes the costs of being a coordinator in a way that preserves network connectivity for a relatively long time. Figures 11 and 12 show results from a single mobile experiment on a 500m×500m area. The 20 source and destination nodes start with 2000 Joules of energy, and the remaining 100 forwarding nodes start with 300 Joules of energy. Without Span, nodes critical to multi-hop routing die around the same time, 335 seconds into the simulation. With Span, the first node failure occurs 615 seconds into the simulation. The packet delivery rate does not drop below 90% until 851 seconds into the simulation, when 12 out of the 100 forwarding nodes are still alive.

Figure 13 shows how much Span extends network lifetime, as a function of node density. We define network lifetime as the time it takes for aggregate delivery rate to drop below 90%. These simulations involve mobility.

Network lifetime with Span is a factor of 2 better than without. Span does not extend network lifetime as much as the energy savings it offers (a factor of over 3.5) because energy consumption rate is not constant. As nodes start to die, the node density decreases. Consequently, the drain rate also increases.

6. Conclusion

This paper presents *Span*, a distributed coordination technique for multi-hop ad hoc wireless networks that reduces energy consumption without significantly diminishing the capacity or connectivity of the network. Span adaptively elects *coordinators* from all nodes in the network, and rotates them in time. Span coordinators stay awake and perform multi-hop packet routing within the ad hoc network, while other nodes remain in power-saving mode and periodically check if they should awaken and become a coordinator.

With Span, each node uses a random backoff delay to decide whether to become a coordinator. This delay is a function of the number of other nodes in the neighborhood that can be bridged using this node, and the amount of energy it has remaining. Our results show that Span not only preserves network connectivity, it also preserves capacity, decreases latency, and provides significant energy savings. For example, for a practical range of node densities and a practical energy model, our simulations show that the system lifetime with Span is more than a factor of two better than without Span.

The amount of energy saving Span provides increases only slightly as density increases. This is largely due to the fact that the current implementation of Span uses the power saving features of 802.11, since nodes periodically wake up and listen for traffic advertisements. Section 5.5 shows that this approach can be extremely expensive. This warrants investigation into a more robust and efficient power saving MAC layer, one that minimizes the amount of time each node in power saving mode must stay up.

7. Acknowledgements

We thank John Ankcorn for providing power measurement hardware and advice.

8. References

[1] Wireless LAN Medium Access Control and Physical Layer Specifications, Aug. 1999. IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee).

[2] Broch, J., Johnson, D., and Maltz, D. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. In *Internet draft, IETF Mobile Ad Hoc Networking Working Group* (Dec. 1998).

[3] Broch, J., Maltz, D., Johnson, D., Hu, Y., and Jetcheva, J. A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Dallas, TX, Aug. 1998).

[4] Chang, J., and Tassiulas, L. Energy Conserving Routing in Wireless Ad Hoc Networks. In *Proceedings of IEEE INFOCOM* (Tel Aviv, Israel, 2000).

[5] Chesson, G. XTP/protocol engine design. In *Proceedings of the IFIP WG6.1/6.4 Workshop* (Rüschlikon, May 1989).

[6] Feeney, L., and Nilsson, M. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc

Networking Environment. In *Proceedings of IEEE INFOCOM* (Anchorage, AK, 2001).

[7] Fenner, W. *Internet Group Management Protocol, Version 2*, Nov 1997. RFC-2236.

[8] Finn, G. G. Routing and Addressing Problems in Large Metropolitan-scale Internetworks. ISI/RR-87-180, ISI, Mar. 1987.

[9] Floyd, S., Jacobson, V., McCanne, S., Liu, C. G., and Zhang, L. A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing. In *Proceedings of the ACM SIGCOMM* (Boston, MA, Sept. 1995), pp. 342–356.

[10] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. Energy-efficient Communication Protocols for Wireless Microsensor Networks. In *Proceedings of the Hawaaian International Conference on Systems Science* (Jan. 2000).

[11] Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Boston, MA, Aug. 2000).

[12] Karp, B., and Kung, H. T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Boston, MA, Aug. 2000).

[13] Kravets, R., and Krishnan, P. Application-driven Power Management for Mobile Communication. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Dallas, TX, Oct. 1998).

[14] Li, J., Jannotti, J., Couto, D. D., Karger, D., and Morris, R. A Scalable Location Service for Geographic Ad-Hoc Routing. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Aug. 2000).

[15] Mangione-Smith, W., Ghang, P. S., Nazareth, S., Lettieri, P., Boring, W., and Jain, R. A Low Power Architecture for Wireless Multimedia Systems: Lessons Learned From Building a Power Hog. In *1996 International Symposium on Low Power Electronics and Design Digest of Technical Papers* (Monterey, CA, Aug. 1996).

[16] CMU Monarch Extensions to `ns`. `http://www.monarch.cs.cmu.edu/`.

[17] ns Notes and Documentation. `http://www.isi.edu/vint/nsnam/`, 2000.

[18] Perkins, C. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the ACM SIGCOMM* (London, U.K., 1994).

[19] Raghavendra, C., and Singh, S. PAMAS: Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks. *ACM Computer Communication Review* (July 1998), 5–26.

[20] Ramanathan, R., and Rosales-Hain, R. Topology Control of Multi-hop Wireless Networks Using Transmit Power Adjustment. In *Proceedings of IEEE INFOCOM* (Tel Aviv, Israel, March 2000).

[21] Rodoplu, V., and Meng, T. H. Minimum Energy Mobile Wireless Networks. In *Proceedings of the IEEE International Conference on Communications (ICC)* (Atlanta, GA, June 1998), vol. 3, pp. 1633–1639.

[22] Rohl, C., Woesner, H., and Wolisz, A. A Short Look on Power Saving Mechanisms in the Wireless LAN Standard Draft IEEE 802.11. In *Proceedings of the the 6th WINLAB Workshop on Third Generation Wireless Systems* (New Brunswick, NJ, Mar. 1997).

[23] Shepard, T. A Channel Access Scheme for Large Dense Packet Radio Networks. In *Proceedings of the ACM SIGCOMM* (Stanford University, CA, Aug. 1996), pp. 219–230.

[24] Singh, S., Woo, M., and Raghavendra, C. S. Power-Aware Routing in Mobile Ad Hoc Networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Dallas, TX, 1998).

[25] Stemm, M., and Katz, R. Reducing Power Consumption of Network Interfaces in Hand-held Devices. In *Proceedings of the Third Workshop on Mobile Multimedia Communications (MoMuC-3)* (Princeton, NJ, 1996).

[26] Wattenhofer, R., Li, L., Bahl, P., and Wang, Y.-M. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *Proceedings of IEEE INFOCOM* (Anchorage, AK, 2001).

[27] Xu, Y., Heidemann, J., and Estrin, D. Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks. Tech. Rep. 527, USC/ISI, Oct. 2000.

[28] Xu, Y., Heidemann, J., and Estrin, D. Geography-informed Energy Conservation for Ad Hoc Routing. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (Rome, Italy, July 2001).