Conserving Energy with On-Demand Topology Management

Cigdem Sengul
Department of Computer Science
Email: sengul@uiuc.edu

Robin Kravets
University of Illinois at Urbana-Champaign
Email: rhk@cs.uiuc.edu

***Abstract*— To reduce idle-time energy consumption, nodes in ad hoc networks can switch to a power-save mode. However, since operating all nodes in power-save mode limits network capacity, some nodes may need to stay in active mode to support forwarding. The main challenge of selecting nodes to stay in active mode stems from the need to conserve energy while maintaining communication. Although topology management protocols build a forwarding backbone of active nodes by powering down redundant nodes, such protocols incur proactive backbone maintenance overhead. The reactive approach, on-demand power management, manages node transitions from active to power-save mode based on routing information. However, node transitions are only traffic-driven and may result in keeping redundant nodes awake. To this end, we propose TITAN, which builds a backbone reactively using information about both ongoing communication and the current power-management mode of nodes. The design of TITAN is based on the trade-offs between waking up power-saving nodes on shorter routes and using longer routes that contain active nodes. Simulation results show that TITAN conserves energy while maintaining efficient communication without additional control overhead for topology management.**

I. Introduction

The disconnected operation of nodes in ad hoc networks requires energy-efficient protocol design to extend node and network lifetime. A promising strategy is to reduce the energy used in wireless communication. While traffic load determines energy used during communication [1], [2], idling energy dissipation dominates system energy consumption in the presence of low to moderate traffic [3], [4]. A common approach to idle-time energy conservation is to switch to a power-save mode where the node is mostly in a sleep state [5]. However, since a node in a sleep state is not capable of communication, a sleep coordination mechanism ensures that nodes that want to exchange traffic are awake at the same time [6], [7]. While power-save mode may support communication in lightly-loaded networks, it imposes additional delay on traffic and severely limits the network capacity as the load increases [4]. To compensate for these limitations, some nodes can stay in active mode (i.e., never power down) and serve as stable relays [3], [4], [8]. Since the choice of active nodes determines both energy consumption and communication quality, the main challenge in idle-time energy conservation is selecting the best set of active nodes through which all traffic flows.

Approaches for selecting active nodes can be categorized into two classes: proactive and reactive. Proactive protocols, known as topology management protocols [3], [8], [9], build a backbone of active nodes, which is typically based on a CDS (Connected Dominating Set), that serves all traffic. However, since the main goal is to preserve network connectivity, the choice of backbone nodes is not tied to network traffic. Therefore, some nodes stay awake even if they are not participating in routing. A reactive approach to this problem is on-demand power management (ODPM) [4], which allows nodes to stay in power-save mode as long as they are not used for routing. Since ODPM simply switches nodes to active if they are on the shortest routes found by the routing protocol, this may result in the unnecessary activation of redundant nodes.

Given the limitations of current approaches to idle-time energy conservation, the contribution of our research is twofold. First, we present an analytical evaluation of potential energy savings from proactive and reactive approaches. We show that proactive approaches save energy by routing through backbone nodes even if such routes are 2-10 times longer than the shortest route that must wake up all nodes on that route. Second, using the insight gained from our analysis, we present an on-demand topology management protocol, TITAN, that combines the benefits of reactive and proactive approaches. From reactive approaches, TITAN allows current network traffic to drive the choice of backbone nodes and, hence, avoids energy consumption due to proactive backbone maintenance. However, as in proactive approaches, once a node is in active mode, it is favored over power-saving nodes for future routes, which enables higher energy savings based on our analysis.

TITAN follows a cross-layer approach by basing the decisions for maintaining a backbone on information from routing and MAC layers. Each node independently decides to join the backbone as route requests flow in the network using its current power-management mode and neighborhood information. Results of extensive simulations show that TITAN provides significant energy savings from eliminating control overhead to build and maintain the backbone. Additionally, due to its on-demand nature, TITAN saves energy by only providing connectivity between active senders and receivers. Furthermore, TITAN achieves high communication performance.

The remainder of the paper is organized as follows. Section II discusses current approaches to idle-time energy conservation. Section III provides an analytical study of energy savings in idle-time energy conservation. Section IV presents the design of TITAN and Section V shows the effectiveness of TITAN via simulations. Finally, Section VI concludes.

II. Idle-Time Energy Conservation in Ad Hoc Networks

The objective of any idle-time energy conservation protocol is to provide energy savings without reducing network capacity. To this end, idle-time energy conservation protocols select a set of active nodes to support communication while other nodes conserve energy in power-save mode. Additional energy savings may also be achieved through transmission power control (TPC) [10]. However, currently, idling costs dominate wireless energy consumption, leaving TPC as a secondary means of saving energy. We are studying the interactions between idle-time energy conservation and TPC as a part of our current work.

Two main challenges for idle-time energy conservation exist. The first challenge is how to select the set of active nodes that handle all traffic. The second challenge is to maintain this set of active nodes without limiting the network lifetime since these nodes may drain their batteries due to forwarding all network traffic. If network lifetime is defined as the time when the first node failure occurs [11], an early death of any node means a short network lifetime. Based on the application, network lifetime may also be defined as the time 1) the fraction of nodes drops below a threshold [3], 2) the aggregate delivery rate drops below a threshold [8], or 3) the first flow dies. Nevertheless, it is essential to balance the energy consumption across all nodes in the network based on the specific definition of lifetime. However, increasing network lifetime is a more difficult challenge since it is dependent on network topology and determined by the minimum lifetime of the cut nodes. Next, we present current proactive and reactive approaches to addressing these challenges in idle-time energy conservation.

A. Proactive Approaches to Idle-Time Energy Conservation

Proactive approaches, such as topology management, address the challenge of selecting active nodes by building a backbone. Such a backbone is typically built based on a CDS in which all nodes are either a member or a direct neighbor of one of the members of the CDS. The backbone preserves total connectivity and serves all traffic in the network [9], [12]. CDS-based protocols provide energy conservation since nodes that are *not* on the backbone can switch to a power-save mode if they are not involved in communication.

To address the challenge of balancing energy usage across all nodes, proactive approaches generally employ *active node rotation* (i.e., nodes resign from the backbone periodically to give other nodes a chance to become active). However, although nodes take turns forming the backbone, some nodes may need to always be active to preserve connectivity. Fig. 1 illustrates a simple case where active node rotation may not avoid overusing nodes. Node 1 has two routes to node 6: $r1$: $1-2-6$ and $r2$: $1-3-6$. Node 2 is a backbone node and forwarding for flow $4 \rightarrow 5$, while node 3 is in power-save mode. However, node 2 has only 50% battery, while the battery of node 3 is full. Nevertheless, active node rotation does not allow node 2 to switch to power-save mode if node 3 cannot take over flow $4 \rightarrow 5$. Even if node 3 can

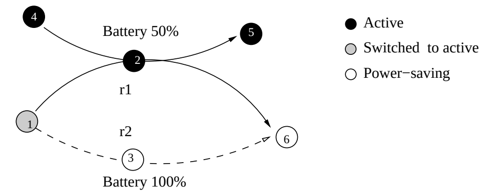

Fig. 1. Example case for active node rotation.

take over this flow, switching node 2 to power-save mode would cause routing disruptions and expensive route discovery for on-demand routing protocols. Therefore, simply rotating active nodes leads to poor energy efficiency when the current communication in the network is not considered.

The main disadvantage of proactive approaches is backbone maintenance, which requires either location information obtained by GPS (e.g., GAF [3]) or topology information obtained by local broadcast messages (e.g., SPAN [8], [9]), both of which consume significant amounts of energy. Broadcast is especially unfriendly to idle-time energy conservation since it keeps power-saving nodes awake. Furthermore, the main goal of CDS-based protocols is to preserve total connectivity in the network. Therefore, although nodes may be removed from the backbone based on node degree and remaining energy [13], more stability in terms of mobility [11] or packet loss characteristics [14], such approaches keep nodes in active mode even if they are not currently participating in forwarding.

B. Reactive Approaches to Idle-Time Energy Conservation

A reactive approach ties power management decisions to network traffic, for instance, information about which nodes are used for routing as in ODPM [4]. The goal is to let nodes that are actively forwarding stay in active mode, while other nodes go into power-save mode. Since a completely connected backbone may not always be necessary (e.g., in an idle network), a reactive approach triggers active node selection in the presence of communication. In ODPM, a node switches to active mode when it receives routing or data packets and starts or refreshes a soft state timer, *keep-alive timer*. On expiration of its keep-alive timer, a node switches from active mode to power-save mode. Since decisions about switching between active and power-save mode are coupled with the routing protocol, which is *not* aware of the power management mode of the nodes along potential routes, ODPM always selects routes independent of other selected routes and may, therefore, activate more nodes than necessary. However, if the communication is well distributed in the network, such independent activation of nodes may provide implicit balancing of energy consumption even though ODPM does not take any explicit measures.

In the next section, we present an analysis of the potential energy savings in idle-time energy consumption based on route selection, which enables a comparison between proactive and reactive approaches and facilitates the design of TITAN.

III. Analytical Study of Energy Savings: Proactive vs. Reactive Approaches

Proactive approaches (e.g., topology management), choose backbone nodes through which all traffic is tunneled, allowing other nodes to stay in power-save mode. However, the average number of hops each packet travels to its destination along the backbone may increase leading to higher communication costs. Reactive approaches (e.g., ODPM), operate based only on network traffic. As traffic flows find shortest routes, the nodes on these routes stay active. Since routes are chosen regardless of the current power-management mode of nodes, redundant nodes may be forced to switch to active, which may increase idling costs. Based on this trade-off, we analyze the conditions for energy-efficiency in routing, which we further use to drive the choice of active nodes. Furthermore, this analysis enables a comparison between proactive and reactive approaches in terms of their potential for energy conservation.

A. Energy-Efficient Routing

The energy-efficiency of routing can be evaluated by comparing the energy consumption of all possible routes for a given flow, ranging from routes where all nodes are already active to routes where all nodes are power-saving and must switch to active. Such a comparison requires an energy model based on the power-management mode of nodes. Therefore, we first derive expressions for energy consumption in active and power-save modes. In our analysis, we assume nodes switch to active if they are forwarding for at least one flow. However, since nodes with low traffic load may still waste energy in idling, we plan to investigate extensions to our analysis that allow such nodes to remain in power-save mode.

The total energy consumed by an active node i, E_A^i, consists of idling costs based on the idling power, P_{idle}, and communication costs based on the reception and transmission powers, P_{rx} and P_{tx}, respectively. The total energy consumed by a power-saving node, E_{PS}, is only the energy spent sleeping, which is determined by the sleep power, P_{sleep}. This model excludes energy costs due to sleep coordination (e.g., synchronization [6]) since all nodes spend some amount of energy in coordination. The energy consumed for sleep/awake transitions is also excluded assuming the wake-up schedule is chosen to amortize these costs. Hence, given a time interval t, (which is composed of idle time, t_{idle}^i, transmission time, t_{tx}^i, and reception time, t_{rx}^i, for an active node i),

$$E_A^i = t_{idle}^i \cdot P_{idle} + t_{rx}^i \cdot P_{rx} + t_{tx}^i \cdot P_{tx}, \tag{1}$$

$$E_{PS} = t \cdot P_{sleep}. \tag{2}$$

The goal of our analysis is to determine if new nodes should switch to active to provide an energy-efficient route for a new flow f, which is defined as follows.

Definition 1: An energy-efficient route imposes the least amount of additional energy cost on the network due to idling, reception and transmission of data and control packets.

The following theorem defines the necessary conditions for selecting an energy-efficient route based on hop count and any necessary transitions to active mode.

Theorem 1: If a relay node spends approximately $c\%$ time in transmission and $c\%$ time in reception additionally for a new flow f, route X is more energy-efficient for flow f than route Y when:

$$(P_{idle} - P_{sleep}) \cdot s_X + \beta_X \leq (P_{idle} - P_{sleep}) \cdot s_Y + \beta_Y, \tag{3}$$

$$\beta_X \approx (P_{tx} + P_{rx} - 2 \cdot P_{idle}) \cdot c \cdot n_X, \tag{4}$$

$$\beta_Y \approx (P_{tx} + P_{rx} - 2 \cdot P_{idle}) \cdot c \cdot n_Y, \tag{5}$$

where n_X and n_Y are the number of nodes on routes X and Y, and s_X and s_Y are the number of power-saving nodes in the respective routes.

Proof: To understand the additional energy cost that a new route imposes on the network, it is essential to differentiate the energy costs of already active nodes and the energy costs of the power-saving nodes that need to switch to active mode. The additional energy consumption of an already active node due to flow f is $\Delta E_{A\to A}^i$. If the times spent in reception and transmission for flow f are $t_{rx_f}^i$ and $t_{tx_f}^i$ respectively, a node that is already in active mode on a given route spends $t_{rx_f}^i + t_{tx_f}^i$ more time in communication and $t_{rx_f}^i + t_{tx_f}^i$ less time in idling. Therefore,

$$\Delta E_{A\to A}^i = -(t_{rx_f}^i + t_{tx_f}^i) \cdot P_{idle} + t_{rx_f}^i \cdot P_{rx} + t_{tx_f}^i \cdot P_{tx}. \tag{6}$$

The energy consumption of a power-saving node that has switched to active mode to forward packets for only flow f is $\Delta E_{PS\to A}^i$. By switching to active mode, an initially power-saving node also spends $t_{rx_f}^i + t_{tx_f}^i$ transmitting and receiving for flow f. However, since this node has incurred no idling but only sleeping costs prior to flow f, the new cost due to flow f includes the time spent idling the remainder of t. Therefore,

$$\begin{aligned}\Delta E_{PS\to A}^i =& (t - (t_{rx_f}^i + t_{tx_f}^i)) \cdot P_{idle} - t \cdot P_{sleep} \\ &+ t_{rx_f}^i \cdot P_{rx} + t_{tx_f}^i \cdot P_{tx}.\end{aligned} \tag{7}$$

Without loss of generality, using (6) and (7), two routes for flow f, routes X and Y, can be compared in terms of energy-efficiency. The energy cost of route X, $E_{cost}(X)$, is the energy consumption from already active nodes and power-saving nodes that have switched to active due to flow f.

$$E_{cost}(X) = \sum_{i=1}^{n_X - s_X} \Delta E_{A\to A}^i + \sum_{i=1}^{s_X} \Delta E_{PS\to A}^i. \tag{8}$$

The additional energy consumed by route Y is $E_{cost}(Y)$. To save energy by choosing route X:

$$E_{cost}(X) \leq E_{cost}(Y). \tag{9}$$

Using (6), (7), and (8), (9) is rewritten as:

$$s_X \cdot t \cdot (P_{idle} - P_{sleep}) + \beta_X \leq s_Y \cdot t \cdot (P_{idle} - P_{sleep}) + \beta_Y, \tag{10}$$

$$\beta_X = \sum_{i=1}^{n_X} -(t_{rx_f}^i + t_{rx_f}^i) \cdot P_{idle} + t_{rx_f}^i \cdot P_{rx} + t_{tx_f}^i \cdot P_{tx}, \tag{11}$$

$$\beta_Y = \sum_{i=1}^{n_Y} -(t_{rx_f}^i + t_{rx_f}^i) \cdot P_{idle} + t_{rx_f}^i \cdot P_{rx} + t_{tx_f}^i \cdot P_{tx}. \tag{12}$$

If each node spends approximately $c\%$ time transmitting and $c\%$ time receiving for flow f, then $t_{rx_f}^i \approx t_{tx_f}^i = c \cdot t$, $0 < c \leq$

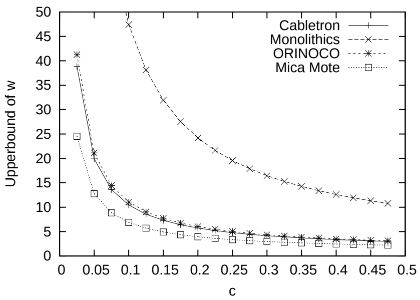

Fig. 2. The upperbound of w as the rate of fbw f increases.

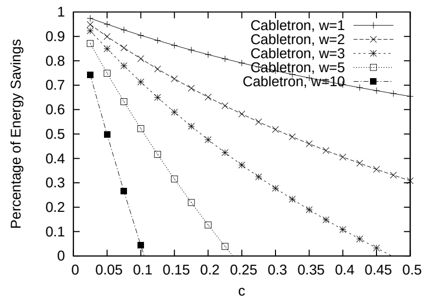

Fig. 3. Percentage of energy savings for Cabletron with $w = 2, 3, 5, 10$.

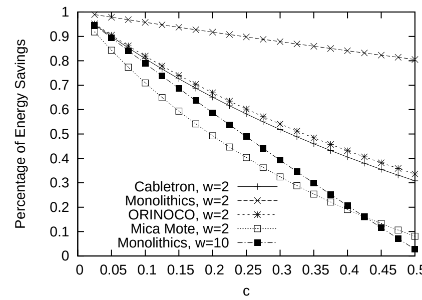

Fig. 4. Percentage of energy savings for different cards with $w = 2, 10$.

TABLE I

ENERGY COSTS (mW) FOR SELECTED WIRELESS CARDS.

Card	P_{tx}	P_{rx}	P_{idle}	P_{sleep}
Cabletron [8]	1400	1000	830	130
Monolithics [15]	14.88	12.50	12.36	0.016
ORINOCO 11b [16]	1400	950	805	60
Mica Mote [17]	81	30	30	0.003

0.5, where $c \cdot t$ includes the time spent for channel contention and retransmissions due to link failures. Therefore, β_X and β_Y can be approximated as:

$$\beta_X \approx c \cdot t \cdot (P_{tx} + P_{rx} - 2 \cdot P_{idle}) \cdot n_X \quad (13)$$

$$\beta_Y \approx c \cdot t \cdot (P_{tx} + P_{rx} - 2 \cdot P_{idle}) \cdot n_Y, \quad (14)$$

which completes the proof of Theorem 1. ■

Theorem 1 provides a rule for choosing routes based on network card energy consumption and c, which is a function of congestion and mobility (i.e., the rate of retransmissions), as well as the rate of flow f. While each node may observe different congestion and mobility, in general, the dominating factor of c is the rate of flow f and therefore, the assumption that all nodes have similar c holds for most cases. Based on this assumption, in the next section, we compare the energy efficiency of proactive and reactive approaches using Theorem 1. However, we are currently investigating the need for a more complex model to account for energy consumption due to congestion and mobility for the cases where this assumption may not hold.

B. Energy Efficiency of Proactive and Reactive Approaches

To compare proactive approaches, which select only active routes, and reactive approaches, which may select routes with all power-saving nodes, we evaluate (3) for two extremes: route X contains only active nodes and route Y contains only power-saving nodes. Hence, using $s_X = 0$ and $s_Y = n_Y$, (3) can be rewritten as:

$$w = \frac{n_X}{n_Y} \leq \frac{c \cdot (P_{tx} + P_{rx} - 2 \cdot P_{idle}) + (P_{idle} - P_{sleep})}{c \cdot (P_{tx} + P_{rx} - 2 \cdot P_{idle})}, \quad (15)$$

which defines a hop count ratio, w, for energy-efficiency when choosing between an active route and a power-saving route. Therefore, for an active route X to provide any energy savings compared to a power-saving route Y, route X must be at most w times longer than route Y. The goal of our study is to understand the upperbound of w that allows energy savings for proactive approaches. We evaluate how much energy savings can be achieved with different w values and if the observed trends hold for different network cards (see Table I).

Since w depends on c (see (15)), Fig. 2 shows the upperbound of w as c increases. For a high c, active routes are more energy-efficient compared to power-saving routes as long as the routing protocol chooses 2-10 times longer active routes. For all cards, the upperbound of w increases as c decreases. In the presence of extremely low traffic, w for each card increases to 25 - 186 (Fig. 2 does not show $w > 100$ for Monolithics at low traffic). Therefore, proactive approaches potentially conserve more energy by directing flows through longer active routes along the backbone compared to reactive approaches, which always choose the shortest route. We quantify the amount of energy savings (i.e., $|E_{cost}(Y) - E_{cost}(X)|/E_{cost}(Y)$) with varying w in Fig. 3. Not surprisingly, the highest energy savings are gained when an active route is the same length as a power-saving route (i.e., $w = 1$). While high energy savings are still guaranteed for all traffic loads when $w = 2$, no energy savings can be provided for high traffic when $w = 3$. Fig. 4 shows these trends hold for all cards. For instance, although Monolithics still provides energy savings when $w = 10$, the energy savings for $w = 2$ are significantly higher. Since the energy savings decrease as w increases, it is essential to bound the route lengths to provide energy savings for all traffic loads. This upperbound is shown to be $w = 2$ in our evaluation, which is representative of the performance with current network cards. However, as the energy characteristics of the cards change, the upperbound of w may need to be reevaluated.

Based on this study, although proactive approaches (i.e., topology management) take the correct approach, they do not consider the impact of route length. Essentially, the energy-efficient routes for each flow may not be present on a backbone based on only connectivity requirements. To this end, we present TITAN, which supports routing through active nodes along a backbone but allows current traffic to trigger backbone formation and maintenance. Essentially, a new node is added to the backbone only if it is on an energy-efficient route.

IV. TITAN: An On-Demand Topology Management Protocol

While a forwarding backbone can be used to support effective communication, the choice of backbone nodes should be driven by the network traffic. Additionally, the overhead for backbone maintenance should be kept to a minimum so that energy savings are not compromised by maintenance costs. To this end, TITAN provides on-demand topology management by using information from both MAC and routing layers. The decisions about backbone nodes are based on our analysis of energy-efficient route selection. The novelty of TITAN comes from its ability to implicitly direct traffic to such energy-efficient routes. Furthermore, TITAN dynamically adapts the backbone to the current network traffic allowing nodes to connect and disconnect from the backbone based on routing decisions. The remainder of this section explains the TITAN protocol in detail.

A. Protocol Design

The main goal of TITAN is to build and maintain an energy-efficient forwarding backbone implicitly and reactively. In contrast to current CDS-based topology management protocols, a backbone in TITAN is defined as follows.

Definition 2: A backbone comprises of nodes that act as a source or a destination or a relay for at least one flow.

Definition 2 ties backbone formation to routing choices in the network, which results in connecting only communicating nodes. To achieve implicit backbone maintenance, each node in TITAN independently decides how to participate in route establishment. Once a route is selected, all nodes along that route join the backbone by switching to active mode. These nodes stay connected to the backbone as long as they continue forwarding for at least one flow.

TITAN is designed to work with an on-demand routing protocol (e.g., DSR [18]) where the source initiates a route discovery by flooding the network with Route Requests (RREQs). While no changes are made to the routing protocol, TITAN impacts a node's decision as to when to forward a RREQ. If there is already a backbone node in a power-saving node's neighborhood, the node defers forwarding the RREQ to allow this node to respond first. Therefore, a power-saving node participates in routing as determined by how long a RREQ is deferred. Assuming the destination sends a Route Reply (RREP) to only the first RREQ, this design ensures that the backbone nodes dominate the route discovery process.

Backbone maintenance in TITAN is realized by three cooperating mechanisms: 1) a *back-off decision mechanism*, 2) a *back-off scheduling mechanism* and 3) *neighbor discovery*. The back-off decision mechanism uses neighborhood information to decide whether or not a power-saving node should back off. The back-off scheduling determines the duration of the back-off for power-saving nodes that decide to back off. Neighbor discovery monitors a node's neighbors to determine their presence and power management mode. Using these mechanisms, TITAN maintains a backbone of active nodes, while relying on ODPM to manage transitions between active and power-save mode. Essentially, ODPM allows traffic-adaptive backbone maintenance since active nodes switch to power-save mode if they are not forwarding for at least one flow. The rest of this section presents these three mechanisms in detail. Additionally, we discuss some solutions for balancing energy consumption, which is not currently supported in TITAN.

B. Back-off Decision Mechanism

In TITAN, each node that receives a RREQ decides independently if it should back off from forwarding the RREQ, which impacts its chance to join the backbone. We formalize the back-off decision mechanism as follows. The neighborhood of a node v_i includes all nodes within its communication range. The degree of v_i is δ_i and α_i is the number of active neighbors of v_i. Since only active nodes act as relays, a forwarding backbone, F, consists of all active nodes in the network. During route discovery, a power-saving node v_i defers sending RREQs to reduce its chance to join F. To reduce the number of active nodes in an area, the back-off decision is tied to δ_i and α_i. Specifically, when δ_i is high, the node redundancy in v_i's neighborhood is high, and therefore, a node should back off with higher probability. If α_i is high, v_i should be more reluctant to join F and so, back off with higher probability, since v_i has neighbors that are already a part of F. Using a simple increasing function of δ_i and α_i, a power-saving node v_i backs off from forwarding a RREQ with probability p_i:

$$p_i = \begin{cases} 1 - \frac{1}{\delta_i^*}, & \text{if } \alpha_i^* = 0 \\ 1 - \frac{1}{\delta_i^* \alpha_i^*}, & \text{otherwise,} \end{cases} \tag{16}$$

where δ_i^* and α_i^* are the number of all neighbors and active neighbors of node i not counting the node that sent the RREQ. While, $\delta_i^* = \delta_i - 1$, α_i^* is not simply $\alpha_i - 1$, since a node that has forwarded a RREQ is not necessarily active. Based on this strategy, the higher (lower) δ_i and α_i, the higher (lower) the probability of back-off. Additionally, if two nodes v_i and v_j have the same α^*, the node with the higher δ^* backs off with higher probability. If v_i and v_j have the same δ^*, the node with the higher α^* backs off with higher probability (see (16)). Therefore, fewer nodes participate in routing in areas with high node density, and in sparse areas, nodes back off with lower probability to reduce the impact of back-offs on route discovery.

C. Back-off Scheduling Mechanism

The back-off scheduling mechanism determines the amount of delay imposed by power-saving nodes that back-off. The duration of the back-off, hence, the delay imposed on a route, impacts if a route is selected by TITAN, since the destination replies to the first RREQ it sees for a particular flow. To determine different possibilities for the back-off duration, it is necessary to understand how power-saving nodes schedule their sleep times. While TITAN in essence is not limited to any sleep coordination mechanism, the discussion in this paper is based on IEEE 802.11 PSM [6]. The reason for our choice is three-fold. First, although not specifically designed for ad hoc networks, IEEE 802.11 PSM is the standard protocol for

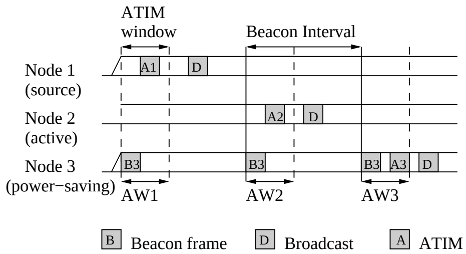

Fig. 5. Back-off scheduling with $d = 1$ using IEEE 802.11 PSM

power management. Second, it has a complete solution for broadcast communication compared to [7]. Third, it does not assume the existence of a wake-up signaling radio as in [19].

In IEEE 802.11 PSM, power-saving nodes are synchronized to wake up at the beginning of every beacon interval. Pending traffic is announced via ATIMs (ad hoc traffic indication messages) in an ATIM window. If a node receives a unicast or a broadcast ATIM, it stays awake for the rest of the interval. Otherwise, nodes switch to sleep state after the ATIM window. To rebroadcast a RREQ, a node should wait for the next ATIM window to guarantee that every node in its neighborhood receives the broadcast (assuming perfect synchronization among nodes).

Due to its synchronized wake-up policy, IEEE 802.11 PSM provides only coarse granularity for back-off scheduling. Therefore, for a normal rebroadcast of a RREQ, a node needs to wait for the next beacon interval. The number of beacon intervals a node backs off from sending a RREQ is denoted as d. Fig. 5 illustrates an example of back-off scheduling with $d = 1$. In the figure, node 1 is the source of the RREQ, node 2 is an active neighbor and node 3 is a power-saving neighbor. Node 1 announces the broadcast in the first ATIM window (AW1). When node 2 receives the broadcast, it sends the broadcast as normal in AW2. Node 3 is in power-save mode and decides to back off (as described in Section IV-B) and announces the broadcast in AW3.

Back-off scheduling exploits the energy-efficiency trade-off between longer active routes and shorter power-saving routes. Our study with current network cards shows that significant energy savings are possible at all traffic loads when an active route is at most twice longer than a power-saving route (i.e., upperbound of $w = 2$, see Section III). Next, we find the value of the d parameter that satisfies this upperbound. This analysis holds only for IEEE 802.11 PSM, since different protocols may enforce different back-off scheduling mechanisms.

In TITAN, sources learn routes that incur the least delay during route discovery. If a RREQ is sent normally the next beacon interval, it incurs a delay of t_N. We denote the delay imposed by a power-saving node that backs off as $t_D \approx (d+1) \cdot t_N$ (since a power-saving node delays a RREQ for d intervals). The average delay incurred on a route r is $delay(r)$

$$delay(r) = \sum_{i=0}^{n_r} (1 - p_i) \cdot t_N + p_i \cdot t_D, \tag{17}$$

where n_r is the number of hops in r. For power-saving nodes, p_i is calculated from (16), while $p_i = 0$ for active nodes. The delay for route set-up for flow f, $delay(f)$, is:

$$delay(f) = \min_{r \in A}\{delay(r)\}, \tag{18}$$

where A is the set of all routes for flow f. The following theorem defines the conditions for route selection in TITAN based on the value of d parameter.

Theorem 2: An active route (i.e., a route containing all active nodes) found by TITAN is at most $d+1$ times longer than a power-saving least-hop route (i.e., the shortest route containing only power-saving nodes).

Proof: Without loss of generality, it is assumed that there are two competing paths for a flow f: route X and route Y. The number of hops on route X and route Y is n_X and n_Y. While route X contains s_X power-saving and a_X active nodes, route Y contains s_Y power-saving nodes and a_Y active nodes. For route X to win the competition (i.e., for the RREQ for flow f to arrive at the destination through route X), the accumulated delay at route X should be less than or equal to the accumulated delay at route Y. Therefore, using (17) and $t_D \approx (d+1) \cdot t_N$, route X may win if:

$$(n_X + \sum_{i=1}^{s_X} p_i \cdot d) \cdot t_N \leq (n_Y + \sum_{i=1}^{s_Y} p_i \cdot d) \cdot t_N. \tag{19}$$

Route X always wins if (19) is a strict inequality. If (19) is an equality then at a certain meeting point, the two routes contend for the channel and the winning route is sent in the RREP. Since $s_X \geq 0$ and $0 \leq p_i \leq 1$, the left side of the equation is greater than or equal to n_X. Additionally, since $s_Y \leq n_Y$ and $0 \leq p_i \leq 1$, the right side of the equation is less than or equal to $(d+1) \cdot n_Y$. Therefore,

$$n_X \leq n_X + \sum_{i=1}^{s_X} p_i \cdot d \leq n_Y + \sum_{i=1}^{s_Y} p_i \cdot d \leq (d+1) \cdot n_Y. \tag{20}$$

■

When $d = 1$, TITAN satisfies the condition $n_X \leq 2 \cdot n_Y$ for selecting energy-efficient routes. Furthermore, the route set-up delay due to back-offs is maintained as at most twice the delay of a least-hop route, since:

$$delay(f) \leq (d+1) \cdot \min_{r \in A}\{n_r\} \cdot t_N = 2 \cdot \min_{r \in A}\{n_r\} \cdot t_N. \tag{21}$$

D. Neighbor Discovery

TITAN assumes that each node knows the number of total and active neighbors (i.e., δ and α). Neighborhood information can be acquired by either periodically sent *hello* messages or snooping transmissions. As in ODPM, to avoid the overhead from *hello* messages, nodes in TITAN keep track of their neighbors by snooping MAC headers, which include the power management mode of the sender. This type of snooping does not incur any additional listening overhead, since a node must try to receive a packet to see if the packet is for itself.

TITAN, as in ODPM, uses a two-stage process for neighbor maintenance. If a packet to an active node fails, it is assumed that the node has switched to power-save mode and the packet

```
FORWARD-RREQ-PSM ()
r_i = UNIFORM-RAND(0, 1)
if α_i^* ≥ 0 and r_i < p_i(*)
   then Buffer RREQ for 1 beacon interval
   else Send RREQ

(*) Calculated from (16)

SEND-BUFFERED-RREQ(rreq)
if CHECK-RREP(rreq.dest → rreq.src) = true
   then Cancel RREQ
   else Send RREQ
```

Fig. 6. Pseudo-code for TITAN.

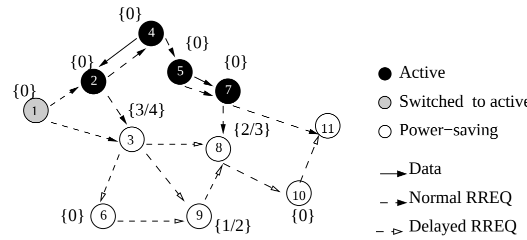

Fig. 7. RREQ propagation when both nodes 3 and 8 back off.

is re-scheduled for the next beacon interval. If the packet fails again, the node is assumed to have moved away. This two-stage process provides a second chance to send to a node that has switched to power-save mode since the last update.

A trade-off with passive discovery is that it does not provide complete neighborhood information. For IEEE 802.11, since nodes periodically send beacon messages for synchronization, even if the network is quiet, passive discovery can still provide some estimation. However, this may not hold for other sleep coordination protocols and a node may be aware of fewer nodes in its neighborhood. While incorrect estimations do not affect the correctness of TITAN and reduce the delay to set up routes, it may reduce the potential for energy conservation.

E. Balancing Energy Consumption in the Network

There is an inherent trade-off between topology management and balancing network energy consumption. TITAN, like all topology management protocols, forces routes through active nodes, which may cause them to die quickly. A naive solution may be to let nodes with low battery delay RREQs longer to reduce the probability of joining the backbone. In the extreme case, a node with limited energy can resign from routing [20]. However, this solution would eventually require a more than necessary number of nodes to stay active, which is counterproductive, since these underloaded nodes consume energy mostly idling and drain their batteries fast.

Although TITAN does not provide explicit energy balancing, it provides implicit balancing as a result of RREQs being delayed by active nodes with long packet queues, which reduces the chance of congested nodes being selected for future flows. This may eventually reduce the load on current active nodes and let them switch to power-save mode faster. Our simulation results also show that TITAN does not have detrimental impact on network lifetime. However, we plan to incorporate explicit energy balancing in TITAN as future work.

F. Illustration of TITAN: An example

TITAN is implemented based on ODPM, IEEE 802.11 and DSR [18]. Power-saving nodes use the distributed algorithm in Fig. 6 for back-off decisions and back-off scheduling. As an optimization, to reduce the number of redundant RREQs in the network, a power-saving node that has heard a RREP for a particular flow cancels sending the buffered RREQ.

Fig. 7 depicts a simple example of backbone maintenance in TITAN. In the example network, nodes 2, 4, 5 and 7 are the active nodes that form the current backbone. When node 1 sends a RREQ for node 11, the RREQ propagation is affected by the RREQ back-off probability assignments of power-saving nodes 3, 6, 8, 9 and 10 (shown in parentheses in Fig. 7). Based on the back-off decisions, TITAN may discover three possible routes for flow $1 \rightarrow 11$: $r1 : 1{-}2{-}4{-}5{-}7{-}11$, $r2(3)$: $1 - 3 - 8 - 7(10) - 11$. TITAN finds $r2$ or $r3$ if both nodes 3 and 8 do not back off. In this case, nodes 3 and 8 join the backbone, while node 10 may join the backbone if it captures the channel before node 7. TITAN finds $r1$ if both nodes 3 and 8 back off, in which case, the backbone stays the same. However, if only node 3 or node 8 backs off, $r1$, $r2$ or $r3$ contend for the channel at a meeting point. The winner of the channel determines if new nodes should join the backbone.

V. PERFORMANCE EVALUATION

The goal of our evaluation is to show the effectiveness of TITAN as an on-demand topology management protocol. To this end, we evaluate TITAN based on its impact on backbone maintenance, energy conservation and communication performance. The metric of interest for characterizing the backbone is the backbone size, which is defined as the average number of active nodes in a unit time interval. The performance in terms of energy is evaluated by energy goodput (bit/J) and lifetime. Energy goodput is defined as the ratio of total application bits received to total energy consumed by data and control (e.g., routing and MAC) packets. Three lifetime definitions are used: 1) the time to the first flow death, 2) the fraction of nodes with non-zero energy as a function of time [3] and 3) the time it takes the aggregate delivery rate to drop below a threshold [8]. We use delivery ratio to measure communication performance, which is the ratio of data packets delivered to data packets sent.

To show TITAN provides implicit topology management, we compare TITAN to ODPM [4] and Span. As a baseline, we also evaluate the performance when all nodes are active (Active). We evaluate SPAN with geographical routing (GR), since the current implementation of Span is coupled with GR. The simulations with Active also use GR, while ODPM and TITAN use DSR. Since GR implementations do not consider any overhead for maintaining location information (e.g., location of the destinations), both Active and Span are at an advantage in terms of routing overhead compared to TITAN and ODPM. Although an exact comparison between

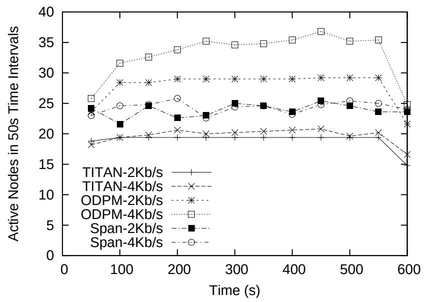

Fig. 8. Number of active nodes vs. time.

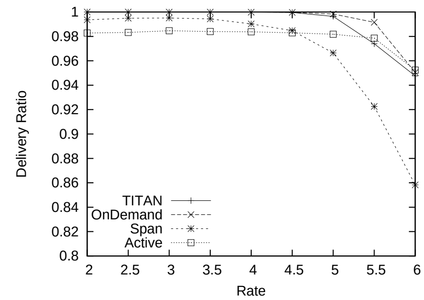

Fig. 9. Delivery Ratio vs. traffic load.

Span and TITAN is not possible due to the use of different routing protocols, the results still provide an understanding of the energy conservation and communication performance of each protocol. Additionally, we evaluate the impact of Span-specific improvements for IEEE 802.11 PSM: 1) individually advertising each broadcast message and 2) using an advertised traffic window so that a node can sleep after it receives all advertised messages. When these options are on, “Opt” is added to the name of the protocol (e.g., Span-Opt). Furthermore, no ATIM messages are sent between active nodes.

We implemented TITAN in ns-2 [21]. In our simulations, we use the Cabletron [8] network card with transmit, receive, idle and sleep powers as 1.4W, 1W, 0.83W and 0.13W respectively. For IEEE 802.11 PSM, the beacon interval is set to 0.3s and the ATIM window is 0.02s as suggested in [8]. When ODPM is used, the keep-alive timers are set to 10s for RREP and 5s for data messages. We simulate two types of networks: 1) Span-topology and static network (described in Section V-A) and 2) uniformly random and mobile network. Our simulation results represent an average of five runs with identical traffic models, but different randomly generated network topologies.

A. Span-Topology Networks

The first set of evaluations follow the Span-topology as described in [8] to enable a fair comparison to Span. In these simulations, 100 forwarding nodes are placed, uniformly at random, in a $1000m \times 1000m$ static network. 10 source and 10 destination nodes are placed, uniformly at random, on each of two 50 meter-wide full-height strips located at the left and right sides of the network. A source on the left side must send to a destination on the right side and vice versa. The traffic is CBR, and the start time for each flow is determined randomly between 20s and 120s. We do not simulate mobility to avoid any control overhead from mobility.

1) Performance Results with Unlimited Energy: To evaluate the backbone maintenance and communication performance of each protocol, we simulate all protocols when nodes have unlimited energy for 600s. We present only the results without Span options for IEEE 802.11 for the sake of brevity since we obtain similar results when these options are turned on.

For the backbone maintenance, based on [8], source and destination nodes are not counted as a part of the backbone. Simulation results show that TITAN uses approximately 20% fewer nodes on average compared to Span, while the difference between TITAN and ODPM is more significant (see Fig. 8). As the traffic load decreases towards the end of the runs, the backbone size in ODPM and TITAN decreases due to active nodes switching to power-save mode as they are no longer required to forward traffic. This behavior is not observed in Span due to its proactive operation. Furthermore, although TITAN does not use three-hop connectivity information that is available to SPAN for backbone maintenance, TITAN’s cross-layer approach allows more intelligent decisions about which nodes need to be active, reducing the backbone size.

Simulation results show that all protocols achieve high delivery ratios (see Fig. 9). Interestingly, for low traffic loads, Span, TITAN and ODPM perform better than Active. While GR using Span coordinators encounters fewer voids and so, achieves a lower loss rate [8], the better results in TITAN and ODPM are an effect of using the two-stage process for maintaining the neighbor table, which gives a second chance to send to a node that has switched to a power-save mode. We also evaluated TITAN’s performance in terms of delay. We omit these results for the sake of brevity but note that TITAN does not incur significant route set-up delays [22].

2) Performance Results with Limited Energy: To evaluate the energy-saving potential of each protocol, we simulate all protocols when all nodes have limited energy. The source and destination nodes start with 2000J of energy, while the remaining 100 nodes start with 300J of energy. Each simulation runs for 1000s. The energy consumption of source or destination nodes is not included in the calculations.

To understand the impact of Span options on energy savings, we evaluate all protocols (except Active) when these options are on and off. Turning these options on in Span compensates for the increase in idling costs due to broadcast messages for backbone maintenance. Broadcast normally keeps nodes awake for the entire beacon interval in IEEE 802.11 PSM. While the energy goodput of TITAN-Opt is 12-68% and 16-30% higher compared to Span-Opt and ODPM-Opt respectively, TITAN achieves 52-130% and 10-30% higher energy goodput compared to Span and ODPM (see Figs. 10 and 11). Additionally, TITAN provides 18-51% higher energy goodput compared to Span-Opt for CBR rates less than 4.5Kb/s, which shows that the energy spent for coordination in Span is

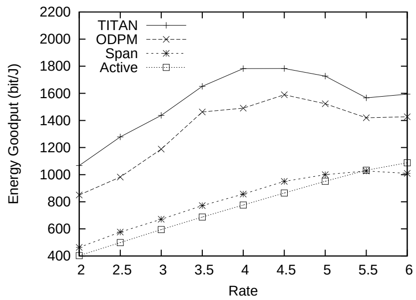

Fig. 10. Energy goodput vs. traffic load without Span options.

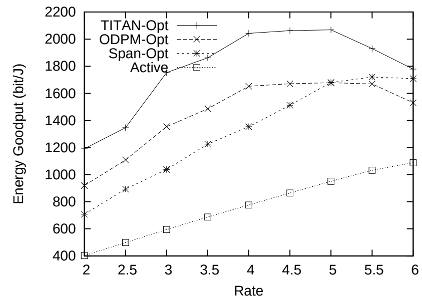

Fig. 11. Energy goodput vs. traffic load with Span options.

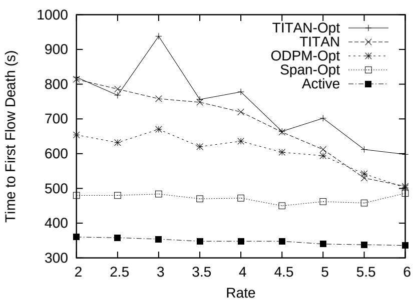

Fig. 12. Time to first flow death vs. traffic load.

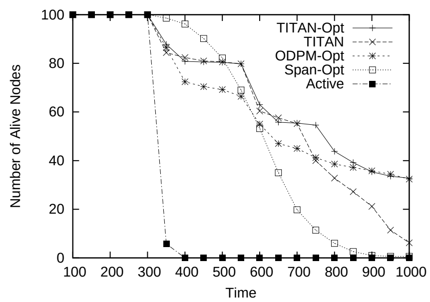

Fig. 13. Number of alive nodes vs. time.

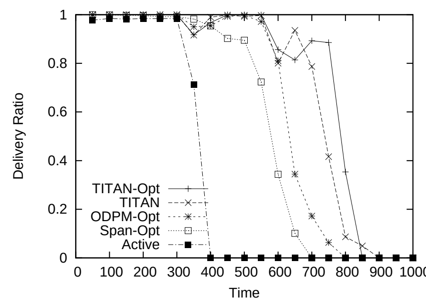

Fig. 14. Aggregate delivery ratio vs. time.

significant even when the options are on. Therefore, even though TITAN-opt achieves higher energy goodput than TITAN, TITAN's performance is not dependent on such options. These results show that TITAN is able to save energy by building and maintaining a backbone only when there is traffic in the network and without any coordination costs.

Next, we studied network lifetime for all protocols with Span options on and also TITAN when Span options are off. We first evaluate the time to first flow death as traffic load increases (see Fig. 12). For TITAN-Opt, the first flow dies 389s later than Active, 266s later than Span, 131s later than ODPM and 56s later than TITAN on average. While TITAN-Opt achieves better performance, TITAN shows more stability. Essentially, using Span options impact the routes found and destabilizes TITAN-opt in terms of the number of backbone nodes and hence, lifetime expectancy, while with TITAN the number of backbone nodes increases with traffic rate, which results in a consistent lifetime behavior. Second, we evaluate the number of alive nodes per unit time when the per flow CBR rate is 4Kb/s (see Fig. 13). Although Span maintains the highest number of alive nodes, between 350s-500s, Span's performance degrades fast. The reason for the fast decrease in the number of alive nodes is due to the active node rotation for energy balancing. For TITAN-Opt and TITAN, the number of alive nodes is more like a staircase function due to focusing traffic to active nodes and switching to other active nodes as former active nodes die. Therefore, the number of alive nodes approaches to 0 more slowly than Span. Finally, we evaluate the aggregate delivery ratio over time when the per flow CBR rate is 4Kb/s (see Fig. 14). TITAN-Opt shows the best performance by maintaining an 88% delivery ratio until 750s, while the delivery ratio falls down to 34% with ODPM-Opt around 650s and with Span-Opt around 600s. For TITAN, the delivery ratio around 600s-650s is 86%. These results show that although TITAN does not provide explicit energy balancing, it does not have a detrimental impact on lifetime.

B. Uniformly Random and Mobile Networks

While static network simulations illustrate the differences between protocols, we simulate more general networks to understand the impact of mobility. In these simulations, nodes are distributed uniformly at random in a $1500m \times 500m$ network. The traffic is CBR, and the start time for each flow is determined randomly between 20s and 25s. We use the extended random waypoint mobility model [23] with node speed uniformly distributed between 1-19m/s and pause times uniformly distributed between 0-120s. Each run is 900s and the steady state average speed is 3.68m/s. All nodes have unlimited energy. To show the different effects of mobility on DSR and GR with perfect location service, we run Active with both DSR (Active-DSR) and GR (Active-GR).

In terms of energy conservation and communication performance, TITAN-opt is able to achieve slightly higher energy goodput and delivery ratios compared to Span and ODPM (see Figs. 15 and 16). Span options improve Span's performance significantly since they reduce the idling energy consumption due to broadcast coordination messages. In a static network, while DSR eventually converges to a set of routes, in a mobile network frequent route breaks lead to expensive route discovery, which amplifies the differences between DSR and

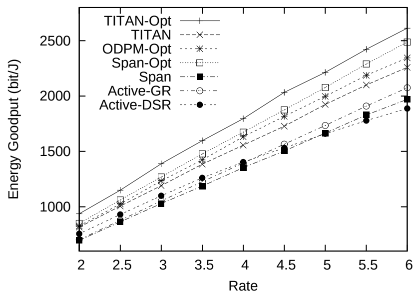

Fig. 15. Energy Goodput vs. traffic load.

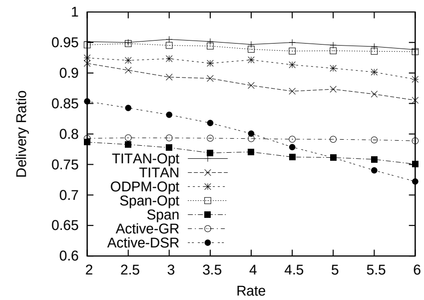

Fig. 16. Delivery ratio vs. traffic load.

GR (e.g., the delivery ratio of DSR decreases significantly as the rate increases in Fig. 16). These results indicate that while TITAN still reduces energy consumption for active communication, the energy consumption due to route discovery becomes more dominant in mobile networks, which requires more efficient route recovery protocols [24], [25].

VI. Conclusion

Conserving energy in ad hoc networks is challenging due to the trade-off between saving energy and maintaining communication. Topology management protocols try to address this challenge by identifying redundant nodes that may stay in power-save mode at the cost of additional control overhead to maintain a backbone of active nodes. To this end, we propose TITAN, which does not require any knowledge of location or coordination among nodes to determine the backbone nodes. Furthermore, due to its on-demand nature, TITAN saves energy by only providing connectivity between active senders and receivers. The results from our analytical and simulation studies show that TITAN achieves efficient topology management without any explicit backbone maintenance and verify that on-demand topology management is the right approach for ad hoc networks. For our future work, we plan to incorporate load balancing into TITAN to achieve fair energy consumption among nodes. Since the design of TITAN is based on our analysis of energy conservation, we will also consider the need for extensions to our energy model to capture dynamic load characteristics in the network.

References

[1] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE INFOCOM*, April 2001.

[2] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, 2000.

[3] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.

[4] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *IEEE INFOCOM*, March 2003.

[5] S. Singh and C. Raghavendra, "PAMAS: Power-aware multi-access protocol with signalling for ad hoc networks," *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, 1998.

[6] IEEE 802 LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard 802.11, 1999.

[7] R. Zheng, J.-C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2003.

[8] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.

[9] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, August 1999.

[10] J. Gomez and A. T. Campbell, "PARO: supporting dynamic power controlled routing in wireless ad hoc networks," *Wireless Networks*, vol. 9, no. 5, pp. 443–460, 2003.

[11] L. Bao and J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2003.

[12] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures," in *IEEE INFOCOM*, April 2001.

[13] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," *KICS Journal of Communications and Networks*, vol. 4, no. 1, pp. 59–70, 2002.

[14] A. Cerpa and D. Estrin, "ASCENT: Adaptive self-configuring sensor networks topologies," in *IEEE INFOCOM*, June 2002.

[15] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Topology management for sensor networks: Exploiting latency and density," in *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002.

[16] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE ad hoc networks," in *International Conference on Parallel Processing*, October 2003.

[17] "Mica2 mote datasheet," www.xbow.com.

[18] D. B. Johnson, D. A. Maltz, and J. Brosch, *Ad Hoc Networking*. Addison-Wesley, 2001, ch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pp. 139–172.

[19] C. F. Chiasserini and R. R. Rao, "Combining paging with dynamic power management," in *IEEE INFOCOM*, April 2001.

[20] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *5th Annual International Conference on Mobile Computing and Networking (MobiCom)*, August 1999.

[21] "Network simulator-ns2 and CMU Monarch extensions to ns-2," http://www.isi.edu/nsnam/ns and http://www.monarch.cs.cmu.edu.

[22] C. Sengul and R. Kravets, "Titan: On-demand topology management in ad hoc networks," UIUC, Tech. Rep. UIUCDCS-R-2004-2481, 2004.

[23] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *IEEE INFOCOM*, March 2003.

[24] C. Sengul and R. Kravets, "Bypass routing: An on-demand local recovery protocol for ad hoc networks," in *The Third Annual Mediterranean Ad Hoc Networking Workshop, (MedHocNet)*, 2004.

[25] M. Spohn and J. J. Garcia-Luna-Aceves, "Neighborhood aware source routing," in *2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2001.