

Trading Latency for Energy in Wireless Ad Hoc Networks using Message Ferrying

Hyewon Jun, Wenrui Zhao, Mostafa H. Ammar, Ellen W. Zegura, and Chungki Lee
College of Computing, Georgia Institute of Technology, Atlanta, Georgia
{hyewon,wrzhao,ammar,ewz,cklee}@cc.gatech.edu

Abstract—

Power management is a critical issue in wireless ad hoc networks where the energy supply is limited. In this paper, we investigate a routing paradigm, *Message Ferrying (MF)*, to save energy while trading off data delivery delay. In MF, special nodes called *ferries* move around the deployment area to deliver messages for nodes. The reliance on the movement of ferries to deliver data increases the delivery delay. However, nodes can save energy by disabling their radios when ferries are far away. To exploit this feature, we present a power management framework, in which nodes switch their power management modes based on the knowledge of ferry location. We evaluate the performance of our scheme using ns-2 simulations and compare it with Dynamic Source Routing (DSR). Our simulation results show that MF achieves energy savings as high as 95% compared to DSR without power management and still delivers more than 98% of data. In contrast, a power-managed DSR delivers much less data than MF to achieve similar energy savings. In the scenario of heavy traffic load, the power-managed DSR delivers less than 20% of data. MF also shows robust performance for highly mobile nodes, while the performance of DSR suffers significantly. Thus, delay tolerant applications should use MF rather than a multihop routing protocol to save energy efficiently when both routing approaches are available.

I. INTRODUCTION

Mobile ad hoc networks (MANETs) consist of wireless nodes that relay data for one another to form a connected network. These networks provide rapid deployment and self configuration capabilities and have applications in a variety of environments such as battlefields, disaster recovery and environmental monitoring. Nodes in MANETs often have limited energy supplies. Thus, to increase the network lifetime, a node should optimize its energy usage. In the system, the wireless interface is one of the largest consumers of energy [12]. The wireless interface consumes energy not only during active communication but also during passive listening, when it is idle. Studies in [6], [20] show that energy consumption while listening to data is only slightly less than it is while actually receiving data. Thus, in the case of moderate traffic load, idle time is the dominating factor in energy consumption and nodes can save considerable energy by “sleeping,” i.e., turning off or disabling their radios, if not used.

In sleeping nodes, data are stored until the nodes wake up. Such nodes can, therefore, achieve energy savings while trad-

ing off data delivery latency. For many applications, latency is not a critical issue. For example, when habitat monitoring nodes collect information periodically and send it to a central node, delivering data ten minutes later does not make much difference. Thus, for these delay-tolerant applications, nodes can save more energy by sleeping longer, while increasing latency. For MANETs using a multihop routing approach ([9], [13], [14], [15]), energy saving techniques that end up trading off latency have been proposed in the literature ([1], [4], [18], [19], [21], [22], [23], [25]). However, there are a number of unresolved problems in techniques that aim to achieve energy savings this way. First of all, sleeping nodes can cause a network to become disconnected, in which data cannot be delivered even if the network is densely deployed. Secondly, if nodes are mobile, the network topology might change during sleeping periods, obsoleting earlier routing information. Reconstructing routing tables or paths would consume additional energy. Finally, accumulating data for a long time and sending them out together increases contention in the network, which results in data loss or additional energy consumption due to retransmission.

In this paper, we consider an alternative routing approach, *Message Ferrying (MF)* [24], and study how it achieves energy savings by trading off latency. In the MF approach, special nodes, called *ferries*, move around the area in which the network is deployed. Similar routing approaches have been proposed for many applications, e.g., ZebraNet [10] to track wild life, DakNet [7] to provide high bandwidth Internet service in rural area, and DataMule [8] to collect data from stationary sensors. The MF approach increases the data delivery delay significantly due to the physical movement of ferries. However, it has important features that enable the network to save energy compared to multihop routing approaches. First, utilizing the knowledge of ferry location, nodes can sleep without degrading performance when ferries are out of communication range. Second, ferries are in charge of data delivery, so nodes do not need to wake up to form a connected network because the ferry mobility eventually connects the network. Also, topology changes in MF do not require any overhead to reconstruct routing tables. Finally, the movement of ferries allows nodes to transmit data at different time according to their locations and decreases contention among nodes. However, these features have not been realized. Zhao et al. in [24] consider that MF is more energy efficient than multihop routing protocols only because it requires less hops to deliver messages.

Chungki Lee is from Myongji University in Korea. This work is done while he is visiting Georgia Institute of Technology.

This work was supported by the NSF Grant ITR-0313062 and AFOSR MURI Grant F49620-00-1-0327.

In this paper, we propose a power management framework for both stationary and mobile nodes. In our framework, nodes switch among different power management modes according to the knowledge of ferry location. We evaluate our schemes using ns-2 simulations and compare them with Dynamic Source Routing (DSR) [9] with and without an idealized power management. Our simulation results show that MF can achieve significant energy savings by trading off latency and still deliver most of the messages. In contrast, power-managed DSR reduces energy consumption at the price of significantly lower delivery rate. For example, MF achieves energy savings up to 95% compared with DSR without power management and delivers over 98% of messages under all traffic loads. However, power-managed DSR delivers as low as 20% of the messages to achieve similar energy savings. In addition, MF shows robust performance for mobile nodes, while the performance of DSR suffers significantly.

The remainder of this paper is structured as follows. In section II, we describe the network model used in our study. Section III presents the power management mechanisms for MF and Section IV shows our simulation results. We describe related work in Section V and conclude the paper in Section VI.

II. NETWORK MODEL

We consider networks consisting of stationary or mobile nodes in a deployment area. Nodes communicate with each other via wireless interfaces. We assume that nodes are identical and are limited in resources. That is, nodes are equipped with the same radios and have the same buffer size and energy supply. In addition, nodes have knowledge of their location and time, e.g., through global positioning system (GPS) or other localization mechanism.

A. Energy Consumption

In this paper, we consider only communication energy consumption and do not account for energy consumption of other sources such as computation. The energy consumption of a wireless interface depends on its activities. Nodes are in one of the five power consumption states, *transmission* state when transmitting, *reception* state when receiving, *idle* state when listening to the wireless medium without transmitting or receiving, *doze* state where the wireless interface is inactive, and *off* state when the wireless interface is turned off and consumes no energy. The amount of energy consumption in each state is assumed based on the studies in [6]. In the doze state, a node consumes an order of magnitude less energy than in the idle state, while a node in the idle state consumes energy at the same order of magnitude as the reception or transmission states. In addition, we consider the transition overhead to turn on the radio, from the off state to the idle state, because it consumes considerable energy.

B. Message Delivery

We consider two approaches for data delivery in the networks, namely *multihop routing* and *message ferrying (MF)*. In

the multihop routing approach, nodes relay messages for one another such that messages can be forwarded from the source to the destination via intermediate nodes. In the MF approach [24], special nodes, called *ferries* move around the deployment area and are responsible for delivering messages for nodes. By carrying messages from the source to the destination, ferries are able to provide communication service to nodes.

In the MF scenario, we consider a network consisting of multiple nodes and a single ferry. We assume that the ferry has ample resources such as large storage and sufficient power supply. To initiate message exchange with nodes, the ferry broadcasts Hello messages, called *beacons*, periodically and nodes in the radio range of the ferry respond to the ferry if they desire to exchange messages. Thus, nodes do not need to form a connected network. Instead, they are required to detect ferry arrival in their neighborhood by listening for beacons and then to exchange messages with the ferry. Figure 1 shows an example of message delivery triggered by beacons in the MF. In this example, the ferry moves down along a central line, while nodes A, B, and C are located beside the line. Initially, the ferry broadcasts a beacon, which is received by nodes A and C. Because it has messages to send to node B, node A sends a response to the ferry, followed by messages. Meanwhile, node C ignores the beacon since it does not have messages to send nor to receive. As it moves down, the ferry keeps broadcasting beacons. Then, node B hears the third beacon. Because the beacon indicates pending messages for itself, node B sends a response to the ferry. After receiving the response, the ferry sends stored messages to node B.

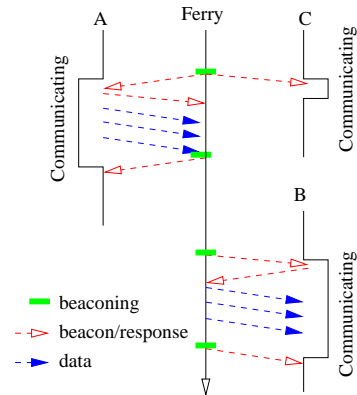


Fig. 1. Message delivery triggered by beacons in Message Ferrying

To specify the movement scenarios, we assume that the ferry is an existing entity such as a shuttle bus. Thus, the movement of the ferry is assumed to be not controllable to assist communication and the ferry does not consume additional energy to move because its movement is required for other purposes. To investigate ideal and practical movement of the ferry, we assume that the ferry moves on a fixed route with either a *strict* schedule or a *loose* schedule and nodes know the route. With a strict schedule, the ferry arrives at each location as it is scheduled. Thus, nodes can estimate when to meet the ferry precisely. With a loose schedule, the ferry is allowed to slow down or pause, which makes it hard to predict the ferry arrival at each location.

III. POWER MANAGEMENT IN MESSAGE FERRYING

A. Power Management Framework

In this section, we describe the framework of our adaptive power saving mechanism. In the mechanism, a node is in one of three power management modes: *sleeping*, *searching*, and *communicating*. In the sleeping mode, a node sleeps because the ferry is out of the communication range. In the searching mode, a node periodically wakes up to listen for a beacon because of insufficient information about ferry movement. Finally, in the communicating mode, a node wakes up frequently to communicate with the ferry in its radio range. To describe the wake-up behavior of a node in each mode, we define three time periods: *wake-up interval*, *beacon period*, and *active window*. A wake-up interval is the time between consecutive wake-up events at a node. A beacon period is the time between consecutive beacon generations by the ferry. Finally, an active window is a fraction of a beacon period, starting from the beginning of a wake-up interval.

These periods are used in the searching or communicating modes as follows. A node wakes up every wake-up interval, which is a multiple of a beacon period. If it does not receive a beacon within an active window, it goes to sleep until the beginning of the next wake-up interval. When a node receives a beacon, if it has any messages to send or to receive, it stays awake for a beacon period. Otherwise, it goes to sleep again until the beginning of the next wake-up interval.

Transitions among the power management modes are triggered by timers or beacon receptions and are shown in Figure 2. Initially, a node estimates the shortest time after which it can communicate with the ferry, called *sleeping time*. Then, it enters the sleeping mode and sets a timer to expire after the sleeping time. When the timer expires, the node estimates its sleeping time, if needed. If it is positive, the node remains in the sleeping mode. Otherwise, the node switches to the searching mode to listen for a beacon. After receiving the first beacon, it switches to the communicating mode. Finally, if the node does not receive a given number of beacons consecutively, it switches to the sleeping mode.

Depending on the movement scenarios, the transition among the modes could be optimized. For example, if the ferry moves on a strict schedule and nodes are stationary, a node can estimate the exact time to communicate with the ferry. Thus, the node may alternate only between the sleeping and communicating modes based on its estimation, without passing through the searching mode. If the ferry moves on a loose schedule and nodes are mobile, a node in the searching mode periodically checks whether it is in the possible range to meet the ferry. If not, it switches back to the sleeping mode instead of keep searching. In addition, when a node in the communication mode detects a ferry departure, it checks whether it is in the possible range to meet the ferry. If so, it switches to the searching mode instead of the sleeping mode.

Figure 3 shows an example scenario in which a node switches its power management mode according to the location of the ferry. A node is in the sleeping mode when the ferry is out of radio range. When it expects to meet the ferry in the near future, it switches to the searching mode and wakes

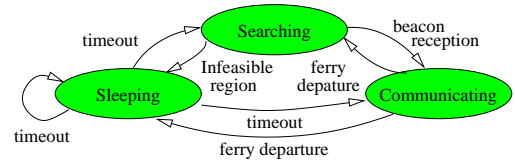


Fig. 2. Transition among power management modes

up periodically to listen for a beacon. After receiving the first beacon, it switches to the communicating mode and frequently wakes up to communicate with the ferry. Finally, when the ferry leaves the radio range, the node switches to the sleeping mode again.

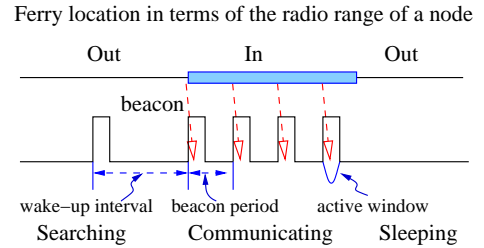


Fig. 3. Power management mode of a node depending on the ferry location

The power management at each mode is designed to save energy based on the characteristics of each mode. Specially, when a node sleeps, it decides whether to sleep in the doze or off state based on the duration of sleeping. If the energy consumption of dozing for the duration is greater than the transition overhead to turn on the radio, a node sleeps in the off state. Otherwise, it sleeps in the doze state. In the sleeping mode, sleeping time is often long because of the physical movement of the ferry. So, a node turns off its radio. The estimation mechanisms of the sleeping time will be described in Sections III-B and III-C. In the searching mode, a node periodically wakes up to listen for a beacon and sleeps if it does not receive a beacon within an active window. The setting of this wake-up interval reflects the trade-off between energy savings and the delivery delay of messages. A longer wake-up interval conserves more energy, but leads to longer delay. Finally, in the communicating mode, a node communicates with the ferry, which is within its radio range. That is, a node wakes up every beacon period to see if it needs to exchange messages with the ferry. In this way, when the ferry receives messages from other nodes destined to this node during the time, the messages can be delivered quickly.

B. Estimation of Sleeping Time for Stationary Nodes

In this section, we explain how to estimate the sleeping time of stationary nodes as well as mobile nodes. To assist the explanation, we use the following notation: A ferry location is represented as $F(t)$ of time t . The route itself is defined as F where $F = \{F(t) | t \geq 0\}$. For a loosely scheduled scenario, $F(t)$ represents the estimated ferry location at t , assuming the ferry moves at its maximum speed. Similarly, a node location at time t is denoted as $N(t)$. If a node is stationary, the location is denoted as N . The maximum speed of the ferry and nodes

are v_F and v_N , respectively. A beacon period is p and the radio radius of nodes and the ferry is r . Finally, the current time is t_0 . Here, we assume t_0 as a multiple of p without loss of generality.

1) *Strictly Scheduled Ferry Movement*: When the ferry moves on a strictly scheduled route, a stationary node can estimate its sleeping time easily by finding when the ferry arrives and leaves its radio range. Figure 4 shows how sleeping time is calculated. Initially, the ferry is at the location of $F(t_0)$ at t_0 , which is in the radio range of a node. The ferry leaves the radio range at t_1 and enters again at t_2 . A node communicates with the ferry until t_1 , when the ferry departs its radio range. Then, the node sleeps until t_2 , when the ferry enters its radio range again. Since a node only needs to wake up in the beginning of a beacon period, the sleeping time estimation requires to be adjusted to reflect the ferry arrival as a discrete time event with time granularity of a beacon period p , starting from time zero.

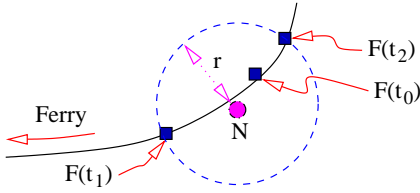


Fig. 4. The intersection of a ferry route and the radio range of a node

2) *Loosely Scheduled Ferry Movement*: This scenario is an extension of the previous scenario. A stationary node estimates the minimum amount of time that the ferry takes to enter the next intersection between the ferry route and the radio range of itself. In fact, the ferry may take longer to enter the intersection because it may slow down or pause in the middle. Thus, a node assumes that the ferry moves at its maximum speed and sleeps only for the minimum amount of time that the ferry takes to enter the next intersection. After sleeping, a node switches to the searching mode and wakes up periodically to listen for a beacon.

C. Estimation of Sleeping Time for Mobile Nodes

In this section, we explain how to estimate the sleeping time of mobile nodes assuming no knowledge of the future movement of the nodes.¹

1) *Strictly Scheduled Ferry Movement*: In this scenario, mobile nodes utilize the precise schedule of the ferry to estimate their sleeping time. When estimating its sleeping time, a node finds the earliest possible time that it can meet the ferry. To calculate this time, a node assumes that it will move directly toward the future location of the ferry at its maximum speed v_N . At time t , if the distance between the future locations of the ferry and the node is greater than r , it is not feasible for the node to be in the radio range of the ferry at t . Thus, the earliest possible time for a node to meet the ferry is the earliest time when the distance between the

¹If a node knows its future movement $N(t)$, the sleeping time can be estimated as if the node were stationary, on the origin of the coordinate, while the ferry moves on $F(t) - N(t)$.

future locations of the ferry and the node becomes less than r . That is, when time is incremented by p , if there exists a minimum non-negative integer k that satisfies

$$|F(t_0 + k \cdot p) - N(t_0)| - v_N \cdot k \cdot p < r, \quad (1)$$

the node will not meet the ferry for a period of $(k - 1)p$. Thus, the node can sleep for $(k - 1)p$. After sleeping, the node determines k from Equation 1 again based on its current location. If k is greater than one, it sleeps again. If k is less than or equal to one, the node switches to the searching mode. In the searching mode, a node calculates k periodically to check whether it has left the radio range of the ferry so that it avoids waiting for a long time in case of losing beacons. If it has departed the radio range, it switches back to the sleeping mode. Otherwise, it stays in the searching mode.

Figure 5 illustrates the above procedure. Currently, a node is located at $N(t_0)$. As time is incremented by units of p , the future locations of the ferry are as follows: $F(t_0 + p)$, $F(t_0 + 2p)$, and so on. Assuming the node moves toward the future location of the ferry, the distance between the future locations is the distance between $F(t)$ and the tip of an arrow, where the length of the arrow represents the distance that a node can move at its maximum speed. Therefore, if the tip of the arrow lies outside of the radio range of the ferry, the node cannot enter the radio range of the ferry by that time. Thus, a node finds the earliest time for the tip of the arrow to enter the radio range of the ferry at its future location and sleeps until right before that time.

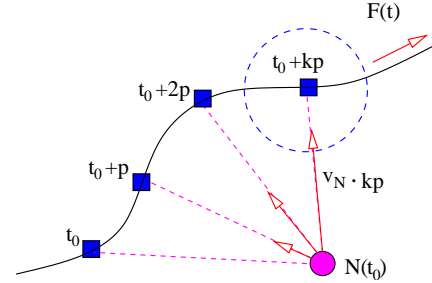


Fig. 5. Sleeping time estimation when a node movement is not known in advance

2) *Loosely Scheduled Ferry Movement*: In this scenario, a node cannot easily estimate when it will encounter the ferry. However, it can estimate when it has no chance encountering the ferry. Clearly, if a node is far away from any location of the ferry route, the node cannot communicate with the ferry. To formulate the problem, denote the distance between a node location $N(t_0)$ and the ferry route F as $d(F, N(t_0)) = \min_t |F(t) - N(t_0)|$. The *feasibility* of a node receiving a beacon is defined as follows:

Definition 1: A node is in the *feasible* radio range of F if the distance between its current location, $N(t_0)$, and the ferry route F is less than a given radio radius r : that is, if

$$d(F, N(t_0)) = \min_t |F(t) - N(t_0)| < r. \quad (2)$$

If a node is in the feasible radio range, it may receive a beacon. Otherwise, it will not receive any beacon. Therefore, estimating the sleeping time of a node is equivalent to finding

the earliest possible time that the node enters the feasible radio range of F .

To estimate the sleeping time, a node assumes that it moves directly toward the closest location of the ferry route at its maximum speed, v_N . Then, the earliest possible time for a node to enter the feasible radio range of F is the earliest time when the distance between the ferry route and the future location of the node becomes less than r . In other words, if there is the minimum positive integer k such as

$$d(F - N(t_0)) - v_N \cdot k \cdot p < r, \quad (3)$$

the node will take at least kp to enter the feasible radio range of F . Thus, the node can sleep for $(k-1)p$. Here k is obtained from the following equation:

$$k = \lceil \frac{d(F - N(t_0)) - r}{v_N \cdot p} \rceil. \quad (4)$$

If k is equal to or less than one, a node switches to the searching mode with a default timeout value. At the timeout, it checks whether it leaves the feasible area using Equation 3.

Figure 6 illustrates an example. A node is currently located at $N(t_0)$. The closest location of the ferry route to $N(t_0)$ is the location where its tangential line and the line connecting the location and a node location intersects at 90 degree. Assuming the node moves toward the intersection, the tip of an arrow is the future location of a node at time kp after moving at its maximum speed v_N . If the tip of the arrow lies outside of the radio range of the intersection at time t , the node cannot enter the feasible radio range by that time. Thus, a node finds the earliest time for the tip of the arrow to enter the radio range from the intersection and sleeps until right before the time.

When a node enters the feasible area, it switches to the searching mode. If a node rarely encounter the ferry, the node can save energy by increasing its wake-up interval without missing the ferry in its radio range most of the time. However, long wake-up interval also decreases the probability that a node will detect the ferry in its radio range. In the next section, we show the trade-off between energy savings and delivery delay with varying wake-up intervals.

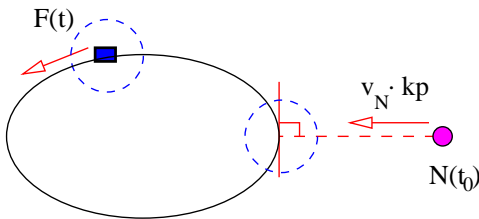


Fig. 6. Checking whether a node is in the feasible radio range of a ferry route

IV. PERFORMANCE EVALUATION

In this section, we demonstrate the trade-off between energy consumption and latency provided by the MF power-management scheme described in Section III. To that end, we use simulations to compare the energy consumption and latency performance of a MANET deploying the MF scheme with one using multihop routing based on the use of Dynamic

Source Routing (DSR) [9]. We choose Dynamic Source Routing (DSR) because it was determined to be the most efficient multihop routing protocol in [3].² In order to provide a fair comparison we consider using DSR in MANETs along with an ideal power management scheme that, while not realizable, provides a bound on the best possible performance of such networks.

The choice of proper power management depends on network topology and the capability of nodes ([1], [4], [18], [19], [21], [22], [25]). Assuming minimal spatial redundancy and no secondary low-power channel, synchronous and asynchronous wake-up mechanisms are the basic wake-up approaches to use.³ Between them, the asynchronous approach is considered to consume more energy than the synchronous approach because nodes usually have to stay awake for a longer time to overlap their awake time with those of their neighbors [25]. Because we are only interested in bounding the performance of MANETs using DSR, we use an idealized synchronous power management scheme. We define three time periods: *wake-up interval*, *awake period*, and *active window*. The wake-up interval is the time between consecutive wake-up events at a node. The awake period is similar to a beacon period in MF and is the time unit for a node to stay awake for message exchange. The active window is a fraction of an awake period, starting from the beginning of a wake-up interval. A node wakes up at the start of a wake-up interval and sends out data or route probing messages, if any. If a node sends or receives any messages within an active window, it stays awake for an awake period to participate in the upcoming communication. Otherwise, it sleeps until the beginning of the next wake-up interval. If it receives any messages during an awake period, it stays awake for another awake period.

A. Simulation Methodology

We use ns-2 simulations to compare the performance of MF and DSR with power management. We also compare them with DSR without power management, called *Continuous Aware Mode (CAM)*. We consider the following four metrics: (1) *energy consumption per node* is defined as an average energy consumption per node in the network, (2) *delivery delay* is defined as an average delay per delivered message, (3) *delivery rate* is defined as the ratio of successfully delivered messages to the total number of generated messages, and (4) *energy cost* is defined as the average energy consumption to deliver a unit message, which is the total energy consumption divided by the number of delivered messages.

In simulations, we use the following default parameters, unless specified otherwise. Our network topology consists of 50 nodes, which are randomly located in a $2000m \times 500m$ region. We use 802.11 MAC and the default power setting of ns-2. For example, the radio range is $250m$ and the data transmission rate is $2Mbps$. Additional power usage

²Other routing protocols, not compared in [3] but used to design power saving mechanisms, tend to have more overhead than DSR in general environments: e.g., geographic routing in [4].

³The power management of MF can also be extended to utilize the spatial redundancy or secondary low-power channel, if they exist. So, we consider basic wake-up mechanisms only as an initial step.

TABLE I

POWER USAGE PARAMETER VALUES USED IN THE SIMULATION (UNIT: W)

State	Transmit	Receive	Idle	Doze	Off
Power	0.2818	0.2053	0.1791	0.0141	0

parameters are shown in Table I. In MF, the ferry uses the same setting as nodes.

To generate traffic, 30 pairs of source and destination nodes are randomly selected and each source chooses a random start time between 10 and 500 seconds. The sources send out messages at a constant rate of one message every 30 seconds for 3000 seconds. We define the traffic load as the total number of bytes generated from all sources in the entire simulation. Each message is of size $1KB$ and has a timeout value of 5000 seconds after which undelivered messages are discarded. Each node has a buffer to store 700 messages, while the ferry has an unlimited buffer space. Each simulation runs for 10,000 seconds and each data point is the average of five runs.

In the implementation of power management, the beacon period in MF and the awake period in DSR are set to 2 seconds and the active window is $500ms$. In addition, we use 10 beacon periods as a wake-up interval in loosely scheduled ferry movement scenarios. Finally, we simulate the energy consumption for a node to turn on its wireless interface. While the amount of consumption depends on devices, the time to resume the radio was measured as $100ms$ for three wireless interfaces in [20]. To assign reasonably large energy consumption, we use $0.05636J$ as the transition overhead to turn on the radio, which is equivalent to the amount of energy to transmit data for $200ms$.

To simulate node movement scenarios, we use the Random Waypoint model [3] as follows. Each node selects a random destination in the region and moves toward the destination at a speed selected randomly between 0 and $10m/s$. When it reaches the destination, it pauses for a *pause time*, which is exponentially distributed with an average of 10 seconds. When the pause time is up, nodes select another random destination to move toward. The ferry moves along a rectangular route, which has (100, 100) and (1900, 400) as two vertices on its diagonal. As a result, the radio range of the ferry swipes through the whole simulation area as the ferry moves along its route. In the scenario of strictly scheduled ferry movement, the ferry moves at a constant speed of $10m/s$. In the scenario of loosely scheduled ferry movement, the ferry moves at a constant speed of $10m/s$ on the edges of the route and pauses at four vertices for a pause time, which is exponentially distributed with an average of 50 seconds.

B. Impact of Traffic Load

In this section, we evaluate the performance of MF and DSR under different traffic loads.

1) *Stationary Nodes*: We first compare the performance of MF and DSR when nodes are stationary. To vary the traffic load, we use message generation intervals of 300, 30, 20, 15, 12, and 10 seconds. In Figure 7, we use DSR- x to represent the case of DSR with power management whose wake-up interval

is x seconds, where x is 2, 50, and 200 seconds. DSR:CAM represents the case of DSR without power management. We also use MF-strict to represent the case of MF with power management where the ferry moves on a strict schedule and MF-loose to represent the case where the ferry moves on a loose schedule.

Figure 7(a) shows the average energy consumption of nodes. Here, MF and DSR with large wake-up intervals (e.g., 50 or 200 seconds) significantly outperform DSR:CAM and DSR-2 under all traffic load. For example, in case of the traffic load of $6MB$, MF and DSR-200 consume only $0.2J$, while DSR-2 and DSR:CAM consume five times or nine times of that, respectively. This is expected because nodes sleep for longer time.

Figure 7(a) also shows that increasing traffic load affects the energy consumption of DSR more than that of MF. In DSR, increasing the number of messages increases the number of transmission multiple times because of relaying the messages. Also, when the power management is used, nodes require to stay awake more to forward more messages. In fact, DSR-2 increases energy consumption faster than DSR:CAM as the traffic load increases, which shows that increasing idle time consumes more energy than increasing transmission by itself in DSR-2.

In Figure 7(b), we show the average delivery delay of messages. The delivery delay of MF is high because of the physical movement of the ferry. In the simulations, the ferry takes at least 420 seconds to come back to the same location. In DSR, using large wake-up interval also increases delivery delay because nodes store messages until the next wake-up interval before relaying if they are asleep. As a result, the delivery delay of MF, DSR-50, and DSR-200 lies in the range of 200 to 600 seconds, while that of DSR:CAM and DSR-2 is under 20 seconds for all traffic load.

Figure 7(c) shows the delivery rate. MF delivers most of the messages regardless of ferry movement scenarios. Meanwhile, DSR delivers fewer messages as the wake-up interval increases because nodes accumulate more messages and send them out at the same time, which increases contention. As the contention level increases, more messages are dropped. Similarly, as traffic load increases, DSR delivers fewer messages due to contention. For example, when the traffic load is $9MB$, DSR delivers only 80, 60, 35, and even 20% of messages if CAM or power management with 2, 50, or 200 wake-up interval is used, respectively. However, MF delivers 98% of the messages under all traffic load.

In addition, Figure 7(c) shows that MF delivers less in a loose schedule scenario than in a strict schedule scenario because of infrequent wake-up in the searching mode. In a loose schedule scenario, a node wakes up only once every ten beacon period in the searching mode. So, it may miss a ferry, which passes through its radio range. If a node keeps missing the ferry and stores messages more than 5000 seconds, the messages are dropped. However, the loss rate is only 2%.

Finally, Figure 7(d) shows the energy cost on a log scale. As the traffic load increases, the energy cost of MF and DSR:CAM decreases because more messages are delivered without increasing energy consumption significantly. In DSR

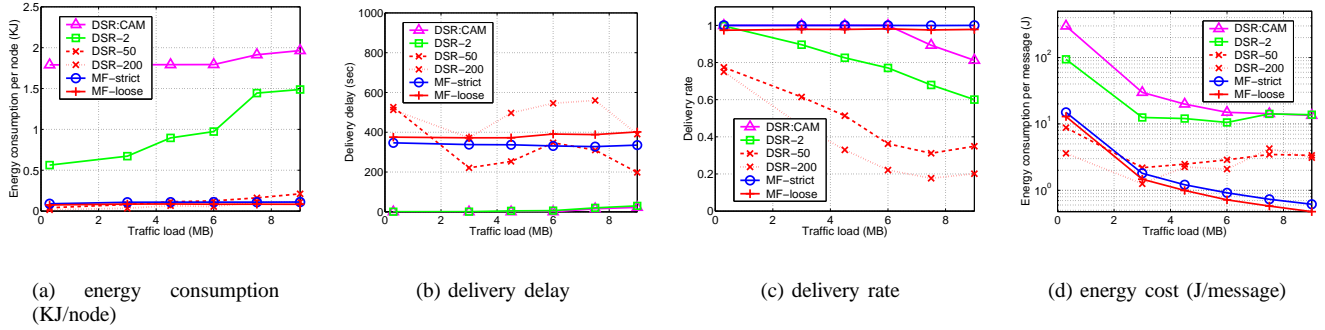


Fig. 7. Impact of traffic loads when nodes are stationary

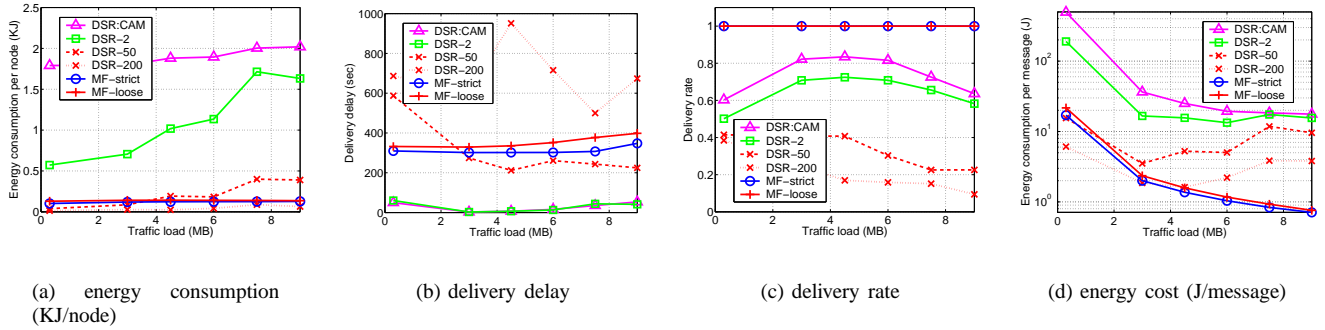


Fig. 8. Impact of traffic loads when nodes are mobile

with power management, when the traffic load is low, energy consumption due to periodic wake-up dominates the total energy consumption. So, energy cost per message decreases as the load increases. When the traffic load is high, more messages are lost due to contention, leading to high energy cost.

2) *Mobile Nodes*: In this section, we evaluate the performance of MF and DSR when nodes are mobile. Many of the results are similar to the stationary node case above. We, therefore, focus on the different features that show up in the simulation experiments.

Figure 8(b) shows the delivery delay. Compared with the stationary node case, MF delivers faster when nodes are mobile because mobile nodes meet the ferry more often. On the other hand, DSR delivers slower when nodes are mobile. Since the node mobility changes topology, nodes are required to probe routing paths before sending out messages if the change occurs. This waiting time for the route probing accounts for the increase in delivery delay.

In Figure 8(c), we show the delivery rate. While MF delivers most of the messages under all traffic loads, the delivery rate of DSR varies significantly. In DSR, a node detects a route change by the failure of message transmission. Thus, it always loses the first message after a route change. In fact, DSR:CAM and DSR-2 have lower delivery rate when traffic load is 300KB than when it is 3MB. Since the first message is dropped after a route change, the former loses a large proportion of messages to discover the route change than the latter. Thus, it has lower delivery rate. Beyond 3MB, the

delivery rate decreases as the traffic load increases due to contention. In case of DSR-50 and DSR-200, both route change and contention decrease the delivery rate significantly. In MF, the node mobility decreases the length of communication time when a node meets the ferry. However, it increases the chance for a node to meet the ferry. Thus, the total communication time between a node and the ferry is not affected much by the node mobility, which results in the steady delivery rate.

C. Impact of Node Mobility

In this section, we evaluate the performance of MF and DSR as node speeds vary from 5 to 50m/s. Because the delivery rate of DSR-50 and DSR-200 is too low, we consider only DSR-2 and DSR:CAM.

Figure 9(a) shows the impact of node speed on the energy consumption. In MF, increasing node speed does not affect the energy consumption of nodes. However, it increases that in DSR because the high node speeds cause more route changes, obsoleting earlier routing information. To reconstruct the routing tables, DSR sends out route probing messages, consuming energy. In addition, if power management is used, nodes stay awake to forward the route probing messages. Due to the latter reason, the energy consumption of DSR-2 increases more than DSR:CAM.

Figure 9(b) shows the impact of node speed on the delivery delay. In MF, the delivery delay decreases as the speed of nodes increases because nodes meet the ferry faster as their speed increases. In DSR, increasing node speed increases the delivery delay of messages because route changes due to high

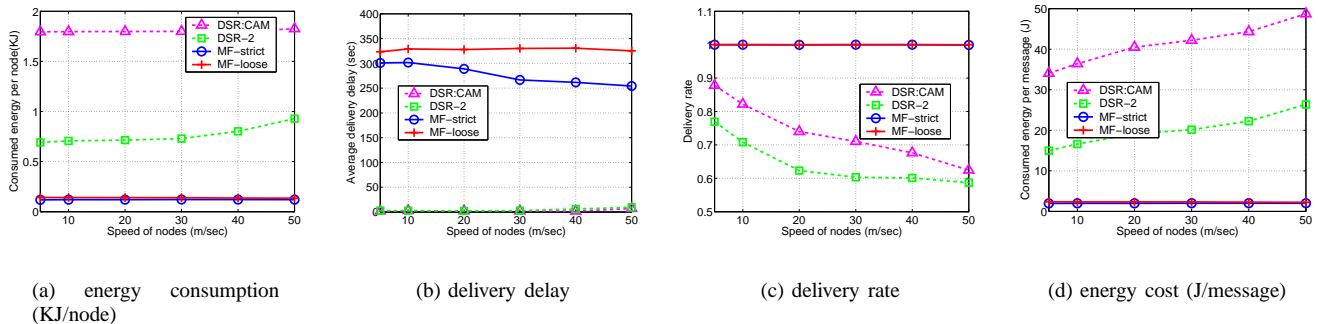


Fig. 9. Impact of node speeds

mobility force nodes to probe routing paths again. Waiting for the results of route probing adds up to the delivery delay. However, it is minor compared with the delivery delay of MF.

Figure 9(c) shows the delivery rate in DSR decreases as the speed of nodes increases because each route change causes the first message to be dropped while detecting the change. However, the node speed does not affect the delivery rate of MF.

Figure 9(d) shows that the energy cost of DSR increases as the speed of nodes increases because more messages are dropped at high speed scenario while more energy is consumed. However, that of MF does not change as the speed increases. Thus, DSR costs more energy to deliver a unit message when the node speeds are high.

D. Impact of Wake-up Interval in Searching Mode

In this section, we evaluate the impact of wake-up interval in the searching mode of MF in a loose schedule scenario. We vary the wake-up interval from two seconds to 200 seconds. We also vary the speed of the ferry as 5, 10, and 25 m/s. In Figures 10 and 11, MF- x m/s represents the case of MF with power management where the maximum speed of the ferry is x m/s.

1) *Stationary nodes*: We first evaluate the performance of MF when nodes are stationary.

Figure 10(a) shows the impact of the wake-up interval on the energy consumption. As the wake-up interval increases, the energy consumption of nodes decreases because nodes sleep more between wake-up events. In fact, the energy consumption is inversely proportional to the wake-up interval.

Figure 10(b) shows that the delivery delay increases as the wake-up interval increases, because a node may miss the ferry by waking up infrequently. If a node misses the ferry that passes through its radio range, the next chance comes after the ferry comes back. Figure 10(b) also shows that the speed of the ferry affects the delivery delay. When the wake-up interval is short, a faster ferry delivers messages faster because it moves faster. However, if the speed of the ferry is high, the ferry stays in the radio range so less time, which increases the probability for a node to miss the ferry. Thus, when the wake-up interval is larger than 20 seconds, the delivery delay of the ferry with the speed of 25 m/s is longer than the case of the slower ferry.

Figure 10(c) shows the impact of the wake-up interval on the delivery rate. The delivery rate decreases as the wake-up interval increases because a larger wake-up interval increases delivery delay. If messages are not delivered within 5000 seconds, they are dropped. Thus, a larger wake-up interval causes more message drops. Figure 10(c) also shows that the delivery rate of faster ferry case is lower than that of slower ferry case. If the ferry moves fast, the delivery delay As the speed of ferry increases, the delivery delay increases. Thus, more messages are dropped due to timeout.

Finally, Figure 10(d) shows that the energy cost decreases as the wake-up interval increases due to energy savings.

2) *Mobile Nodes*: We now evaluate the performance of MF when nodes are mobile. Figures 11(a) and (b) show the trade-off between energy and delay similarly to Figure 10. The degree of energy savings is greater than that of stationary nodes because mobile nodes spend more time in the searching mode than stationary nodes. Also, Figure 11(c) shows that MF has a better delivery rate than DSR under all three speeds of the ferry even when the wake-up interval is as large as 100 seconds. Because the ferry swipes through the entire region, the ferry provides better network connection than DSR when nodes are mobile. Finally, Figure 11(d) shows that the energy cost decreases inversely with the wake-up interval increases.

In summary, both stationary and mobile nodes can save energy by increasing wake-up interval in the searching mode. However, a very large wake-up interval reduces the delivery rate. Therefore, the wake-up interval that is less than 50 seconds is appropriate in these scenarios.

V. RELATED WORK

In this section, we review some related work on power management in wireless networks such as cycling between sleeping and waking up, using mobility, tuning power management modes according to traffic patterns, and trading latency for energy.

Wake-up mechanisms have been developed based on measurement studies [6], [20], which show that energy consumption while listening to data is almost as high as that while actually receiving data. Thus, nodes can save considering energy by “sleeping,” i.e., turning off or disabling their radios, if not used. In multihop ad hoc networks, wake-up mechanisms fall into four categories: *synchronous*, *asynchronous*, *cell-based*,

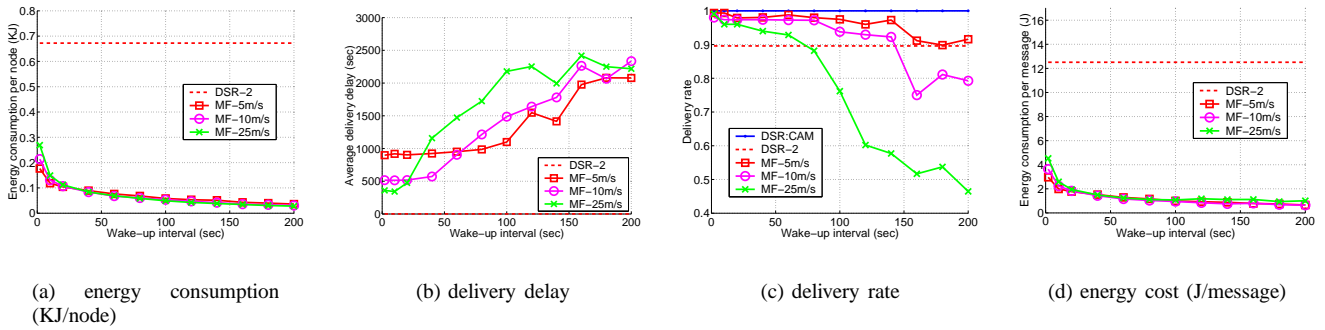


Fig. 10. Impact of wake-up intervals in the searching mode where nodes are stationary and the ferry moves on a loose scheduled route

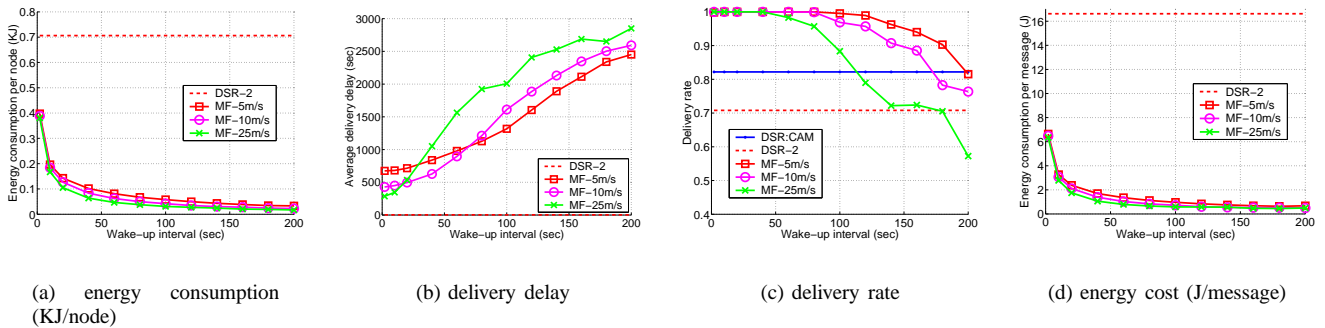


Fig. 11. Impact of wake-up intervals in the searching mode where nodes are mobile and the ferry moves on a loose scheduled route.

and *on-demand mechanism*. In synchronous mechanisms such as 802.11 Power Saving Mode(PSM) [1], nodes wake up periodically and notify pending messages to intended receivers in order to make them stay awake. However, 802.11 PSM is designed for a fully connected network and not suitable for sparse or partitioned networks. In asynchronous mechanisms [21], [25], neighboring nodes wake up in the way their awake time interval overlapped one another and connect a network eventually without aid of clock synchronization among nodes. In this approach, the shortage of time information forces nodes to stay awake longer than in synchronous mechanisms. In cell-based mechanisms [4], [22], the deployment area is divided into cells and a few coordinators in each cell are elected to connect the network while others sleep. This approach utilizes the spatial redundancy in densely deployed networks. Finally, in on-demand approaches [19], [18], nodes are assumed to have secondary low-power channels to connect a network and wake up the main communication channel, if needed. This approach utilizes the hierarchical architecture of devices to save energy. Our power management approach in MF utilizes the information of ferry locations to determine when to switch between sleeping and waking up.

The design of power management is greatly affected by routing protocols in networks. In highly partitioned networks, nodes deliver messages using mobility of their own or others [10], [7], [17], [24], [5]. Jain et al. [17] utilize this feature to save energy while collecting data from stationary sensors. They use short range radio to reduce energy consumption and vary the duty cycle of sensors. However, they didn't

utilize the location information of the mobile nodes to save energy. Also, the energy consumption in the idle state was not considered. In this paper, we show that utilizing the location information of mobile nodes helps to minimize the idle energy consumption.

Traffic patterns are also important factors in the design of power management mechanisms. Zheng et al. [26] presented on-demand power management, in which a node switches its power management modes between Continuous Aware Mode and Power Saving Mode according to incoming data traffics in wireless LAN using 802.11 MAC protocol. They observe that once a packet arrives, more packets tend to follow and form a flow. Thus, if a node receives a packet, it stays awake to increase throughput and decrease latency for the duration of a flow. Anand et al. [2] also proposed to tune power management modes adaptively to the application and network characteristics. Our approach also switches among different power states according to traffic patterns.

Trading latency for energy has been investigated based on the modulation scaling theory. According to the theory, the transmission power is proportional to the transmission rate. Therefore, sending a packet slowly using low transmission power can save energy. However, packets cannot be transmitted arbitrarily slowly on a shared medium. As a result, various scheduling algorithms, called *lazy scheduling*, have been proposed to transmit packets slowly within given constraints, e.g., a latency bound in [16], [11], [23]. The power management mechanisms in MF trade between latency and energy according to different network characteristics.

VI. CONCLUSIONS

In this paper, we investigate the use of the Message Ferrying routing scheme to save energy while trading off delay. We present a power management framework, in which nodes switch among different power management modes according to the knowledge of ferry location. Using ns-2 simulations, we evaluate the performance of MF and compare it with DSR. Our simulation results show that MF can achieve significant energy savings by trading off latency, while still delivering most of the messages. In contrast, power-managed DSR reduces energy consumption at the price of significantly lower delivery rate. In addition, MF shows robust performance in the face of node mobility.

REFERENCES

- [1] Wireless LAN medium access control and physical layer specifications, August 1999. IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee).
- [2] M. Anand, E. B. Nightingale, and J. Flinn. Self-tuning wireless network power management. In *MobiCom*, 2003.
- [3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobicom*, 1998.
- [4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *MobiCom*, 2001.
- [5] K. Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM*, 2003.
- [6] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc network. In *INFOCOM*, 2001.
- [7] A. A. Hasson, R. Fletcher, and A. Pentland. DakNet: A road to universal broadband connectivity. Wireless Internet UN ICT Conference Case Study, 2003. <http://www.medialabasia.org/>.
- [8] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy efficient data collection in sensor networks. In *Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt)*, 2004.
- [9] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, 1996.
- [10] P. Juang, H. Oki, and Y. Wang. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *ASPLoS-X*, San Jose, CA, October 2002.
- [11] R. R. Kompella and A. C. Snoeren. Practical lazy scheduling in sensor networks. In *Sensys*, 2003.
- [12] R. H. Kravets. Cooperative solutions to the dynamic management of communication resources. In *Ph.D. thesis*, 1998.
- [13] V. D. Park and M. S. Corson. Highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM*, 1997.
- [14] C. E. Perkins. Ad-hoc on-demand distance vector routing. In *Presentation in MobiCom*, 1999.
- [15] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM*, 1994.
- [16] B. Prabhakar, E. U. Bıyıkoglu, and A. E. Gamal. Energy-efficient transmission over a wireless link via lazy packet scheduling. In *INFOCOM*, 2001.
- [17] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a three-tier architecture for sparse sensor networks. In *IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.
- [18] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: An event driven energy saving strategy for battery operated devices. In *MobiCom*, 2002.
- [19] C. Shurgers, V. Tsatsis, S. Ganeriwai, and M. Srivastava. Topology management for sensor networks: Exploring latency and density. In *MobiHoc*, 2002.
- [20] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, E80-B(8):1125–1131, August 1997.
- [21] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for IEEE 802-11-based multi-hop ad hoc networks. In *INFOCOM*, 2002.
- [22] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom*, 2001.
- [23] Y. Yu, B. Krishnamachari, and V. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. In *INFOCOM*, 2004.
- [24] W. Zhao, M. H. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc*, 2004.
- [25] R. Zheng and R. Kravets. Asynchronous wakeup for ad hoc networks. In *MobiHoc*, 2003.
- [26] R. Zheng and R. Kravets. On-demand power management for ad hoc network. In *INFOCOM*, 2003.