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, power-managed DSR delivers much less data than MF to achieve similar energy savings. In the scenario of heavy traffic load, 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.*

1 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 [10]. The wireless interface consumes energy not only during active communication but also during passive listening, when it is idle. Studies in [4, 16] 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 trading 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 [7, 11, 12, 13], energy saving techniques that end up trading off latency have been proposed in the literature [1, 3, 14, 15, 17, 18, 19, 21]. 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)* [20], 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.,

---

[*] 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.

ZebraNet [8] to track wild life, DakNet [5] to provide high bandwidth Internet service in rural area, and DataMule [6] 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 [20] consider that MF is more energy efficient than multihop routing protocols only because it requires less hops to deliver messages.

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) [7] 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 2, we describe the network model used in our study. Section 3 presents the power management mechanisms for MF. In Section 4, we show our simulation results. We conclude the paper in Section 5.

2 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.

2.1 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 [4]. 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.

2.2 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 [20], 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.

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.

3 Power Management In Message Ferrying

3.1 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 1. 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. More complicated scenarios are discussed in a more detailed technical report [9].

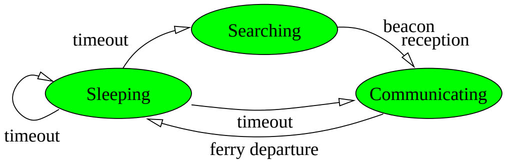


Figure 1. Transition among power management modes

Figure 2 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 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.

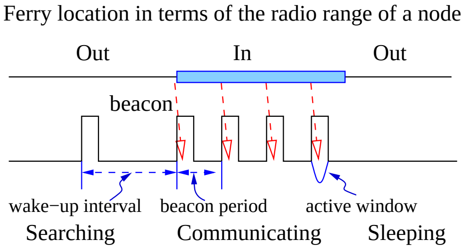


Figure 2. Power management mode of a node depending on the ferry location

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 Section 3.2. 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.

3.2 Estimation of Sleeping Time

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 and a node location at time t are represented as $F(t)$

and $N(t)$, respectively. The maximum speed of the ferry and nodes are v_F and v_N. 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.

3.2.1 Stationary Nodes

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.

When the ferry moves on a loosely scheduled ferry route, a stationary node estimates the minimum amount of time that the ferry takes to enter its radio range again. 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.

3.2.2 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] In the scenario where the ferry moves on a strictly scheduled route, a mobile node can estimate its sleeping time by finding 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 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

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

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 3 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.

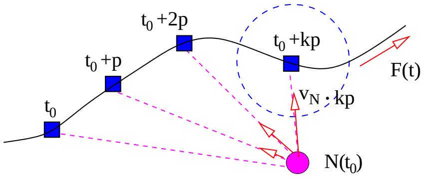


Figure 3. Sleeping time estimation when a node movement is not known in advance

In the scenario where the ferry moves on a loosely scheduled route, mobile nodes estimate its sleeping time by finding the earliest possible time to meet the ferry. This is discussed in the more detailed technical report [9].

4 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 3.[2] 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 Dynamic Source Routing (DSR) [7]. We choose Dynamic Source Routing (DSR) because it was determined to be the most efficient multihop routing protocol in [2].[3] 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, 3, 14, 15, 17, 18, 21]. Assuming minimal spatial redundancy and no secondary low-power channel, synchronous and asynchronous wake-up mechanisms are the basic wake-up approaches to use.[4] Between them, the asynchronous ap-

---

[1] If 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 $F(t) - N(t)$.

[2] Additional simulation results can be found in [9].

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

[4] The 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.

proach 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 [21]. 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.

4.1 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 three metrics: (1) *delivery delay* is defined as an average delay per delivered message, (2) *delivery rate* is defined as the ratio of successfully delivered messages to the total number of generated messages, and (3) *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. The detailed simulation settings are described in the technical report [9].

4.2 Impact of Traffic Load

In this section, we evaluate the performance of MF and DSR under different traffic loads. To vary the traffic load, we use message generation intervals of 300, 30, 20, 15, 12, and 10 seconds. In Figure 4, 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.

We first compare the performance of MF and DSR when nodes are stationary. In Figure 4(a), 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.

Figure 4(b) 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 the traffic load increases, DSR delivers fewer messages due to contention.

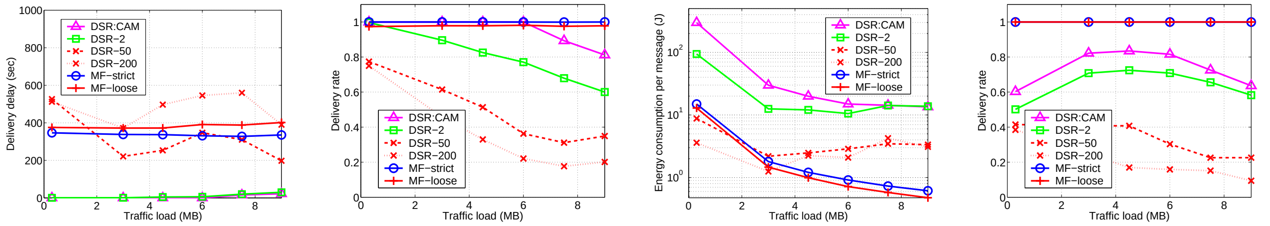
Figure 4(c) shows the energy cost on a log scale. MF and DSR with large wake-up intervals (e.g., 50 or 200 seconds) significantly outperform DSR:CAM and DSR-2 under all traffic loads, because nodes sleep for longer time. For example, in case of traffic load of $3MB$, MF and DSR-200 consume less than $2J$ per delivered message, while DSR-2 and and DSR:CAM consume more than six times or 15 times of that energy, respectively. As the traffic load increases, the energy cost of MF decreases because more messages are delivered without increasing energy consumption significantly. In DSR 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.

Figure 4(d) shows the delivery rate for the case of mobile nodes. 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 node mobility, which results in the almost constant delivery rate.

4.3 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 rates of DSR-50 and DSR-200 are too low, we consider only DSR-2 and DSR:CAM, where the message generation interval is 30 seconds.

Figure 5 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.



(a) delivery delay for the case of stationary nodes

(b) delivery rate for the case of stationary nodes

(c) energy cost for the case of stationary nodes

(d) delivery rate for the case of mobile nodes

Figure 4. Impact of traffic loads for stationary and mobile nodes

However, the node speed does not affect the delivery rate of MF.

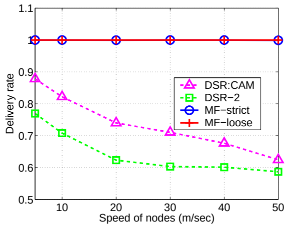


Figure 5. Impact of node speeds

5 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] 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.

[3] 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.

[4] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc network. In *INFOCOM*, 2001.

[5] 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/.

[6] 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.

[7] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, 1996.

[8] 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.

[9] H. Jun, W. Zhao, M. H. Ammar, E. W. Zegura, and C. Lee. Trading off latency for energy in wireless ad hoc networks by message ferrying. Technical Report GIT-CC-04-13, Georgia Institute of Technology, 2004.

[10] R. H. Kravets. Cooperative solutions to the dynamic management of communication resources. In *Ph.D. thesis*, 1998.

[11] V. D. Park and M. S. Corson. Highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM*, 1997.

[12] C. E. Perkins. Ad-hoc on-demand distance vector routing. In *Presentation in MobiCom*, 1999.

[13] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM*, 1994.

[14] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: An event driven energy saving strategy for battery operated devices. In *MobiCom*, 2002.

[15] C. Shurgers, V. Tsiatsis, S. Ganeriwai, and M. Srivastava. Topology management for sensor networks: Exploring latency and density. In *MobiHoc*, 2002.

[16] 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.

[17] 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.

[18] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom*, 2001.

[19] Y. Yu, B. Krishnamachari, and V. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. In *INFOCOM*, 2004.

[20] W. Zhao, M. H. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. MobiHoc, 2004.

[21] R. Zheng and R. Kravets. Asynchronous wakeup for ad hoc networks. In *MobiHoc*, 2003.