ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT

Ad Hoc Networks xxx (2006) xxx–xxx

Ad Hoc Networks

www.elsevier.com/locate/adhoc

Trading latency for energy in densely deployed wireless ad hoc networks using message ferrying ☆

H. Jun *, W. Zhao, M.H. Ammar, E.W. Zegura, C. Lee [1]

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, United States

Received 4 October 2005; received in revised form 30 January 2006; accepted 6 February 2006

Abstract

Wireless mobile ad hoc networks (MANETs) have the potential for use in important application environments, such as remote environmental monitoring, where energy resources are limited. Efficient power management is necessary to allow these networks to operate over a long period of time. One of the key factors affecting the design of power management mechanisms is the routing protocol in use within the network. In this paper, we investigate the *Message ferrying* (*MF*) routing paradigm as a means 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. While this routing paradigm has been developed mainly to deliver messages in partitioned networks, here we explore its use in a connected MANET. The reliance on the movement of ferries to deliver messages increases the delivery delay if a network is not partitioned. However, delegating message delivery to ferries provides the opportunity for nodes to save energy by aggressively 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 knowledge of ferry location. We evaluate the performance of our scheme using ns-2 simulations and compare it with a multihop routing protocol, 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 messages. In contrast, a power-managed DSR delivers many fewer messages than MF to achieve similar energy savings. In the scenario of heavy traffic load, the power-managed DSR delivers less than 20% of messages. MF also shows robust performance for highly mobile nodes, while the performance of DSR suffers significantly. Thus, delay tolerant applications can use MF rather than a multihop routing protocol to save energy efficiently when both routing approaches are available.

Keywords: Power management; Message ferrying; Multihop routing protocols; Mobile ad hoc networks

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

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

* Corresponding author. Tel.: +1 404 723 4226.
E-mail address: hyewon@cc.gatech.edu (H. Jun).
[1] C. Lee is from Myongji University in Korea. This work was done while he was visiting Georgia Institute of Technology.

1570-8705/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2006.02.001

such as battlefields, disaster recovery, and environmental monitoring. However, nodes in MANETs often have limited energy supplies. Therefore, efficient power management mechanisms are necessary to allow these networks to operate over a long period of time. In energy-limited devices, the wireless interface is one of the largest consumers of energy [1]. In addition to consuming energy during active communication, the wireless interface also consumes a significant amount of energy in the idle mode while listening for transmissions by other nodes. Studies have shown that energy consumption while listening to data is almost as high as that while actually receiving data [2,3]. 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 communicating.

In sleeping nodes, data is stored until the nodes wake up. Such nodes can, therefore, achieve energy savings while trading off data delivery latency. For some 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 10 min 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 [4–7], energy saving techniques that end up trading off latency have been proposed in the literature [8–15]. 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, and as a consequence data cannot be delivered even if the network is densely deployed. Secondly, if nodes are mobile, the network topology might change during sleeping periods, making earlier routing information obsolete. 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*) [16,17]. In the MF approach, special nodes called *ferries* move around the area in which a network is deployed. These ferries are in charge of delivering messages among nodes as shown in Fig. 1. When node A has a message for node B, it sends the message to a ferry when they are close to each other. Then, the ferry moves along its planned route. When the ferry becomes close to node B, it sends the message to node B. Similar routing approaches have been proposed for many applications, e.g., ZebraNet [18] to track wild life, DakNet [19] to provide high bandwidth Internet service in rural areas, and DataMule [20] to collect data from stationary sensors. These routing approaches have been developed mainly to deliver messages in sparsely deployed and partitioned networks.

In this paper, we consider the use of MF in a network with *densely deployed nodes*, and study how to achieve energy savings by trading off latency. The use of MF can increase data delivery delay over "traditional" MANET multiple routing protocols (e.g., DSR [4], AODV [6], and DSDV [7]). However, it has important features that enable the network to save energy compared to these 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

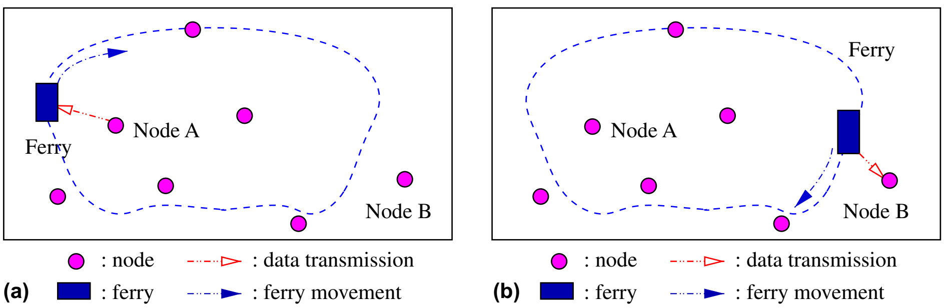

Fig. 1. An example of message delivery from node A to node B using MF. (a) From node A to Ferry. (b) From Ferry to node B.

in MF do not require any overhead to reconstruct routing tables. Finally, the movement of ferries allows nodes to transmit data at different times according to their locations and decreases contention among nodes.

To exploit these features of MF, 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) [4] with and without an idealized power management scheme. Our simulation results show that MF can achieve significant energy savings. 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 and Section 4 shows our simulation results. We describe related work in Section 5 and conclude the paper in Section 6.

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 mechanisms.

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 or mobility. The energy consumption of a wireless interface depends on its activities, i.e., *transmitting*, *receiving*, *idling* (when listening to the wireless medium without transmitting nor receiving), *dozing* (when the wireless interface is inactive), and being *off* (when the wireless interface is turned off and consumes no energy). The amount of energy consumption in each activity is assumed based on the studies in [2]. When dozing, a node consumes an order of magnitude less energy than when idling, while an idling node consumes energy at the same order of magnitude as a receiving or transmitting node. Table 1 shows the power usage parameter values used in our simulations. In addition, we consider the transition overhead to turn on the radio, from being off to idling, because it consumes considerable energy.

Table 1
Power usage parameter values used in the simulation (unit: W)

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

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 [17], 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.[2] We assume that the ferry has ample resources such as large storage and sufficient power supply.[3] 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

[2] When a network has multiple ferries, more issues arise, such as dealing with coordination and different movement patterns of ferries [21]. Thus, we leave the consideration of multiple ferries to future work.

[3] In the multihop routing scenario, it does not help that much for a network to have such a node because the other nodes consume as much energy as before regardless of the additional resources and limit the network lifetime.

ARTICLE IN PRESS

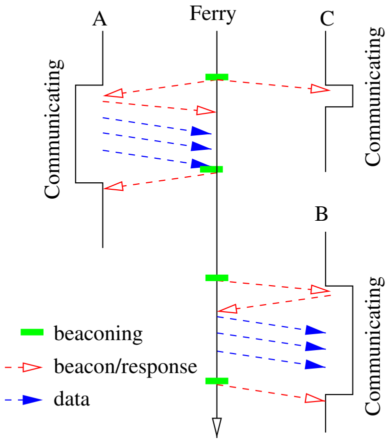

Fig. 2. Message delivery triggered by beacons in message ferrying.

network. Instead, they are required to detect ferry arrival in their neighborhood by listening for beacons and then to exchange messages with the ferry. Fig. 2 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.

To specify the movement scenarios, we assume that the ferry is an existing entity whose movement is assumed not controllable for the purpose of assisting communication and is required for other purposes.[4] 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 in which 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 (i.e., dozes or turns off its radio) 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 Fig. 3. 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

[4] As an example of the MF approach, a shuttle bus in a national or amusement park can be used as a ferry to collect information from sensors deployed in the park. Since the movement of the shuttle bus is needed for transportation, the energy for the ferry movement does not need to be considered. Also, the energy for the communication part of the ferry is easily rechargeable in such a vehicle, so it does not limit the lifetime of the network.

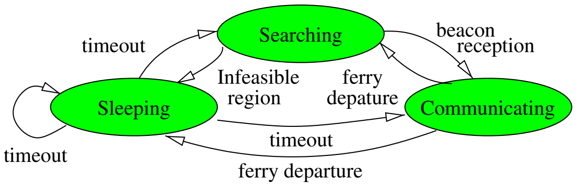

Fig. 3. Transition among power management modes.

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, in the case that 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. In the case that the ferry moves on a loose schedule and nodes are mobile, at the ferry departure, a node may switch from the communicating mode to the searching mode, instead of the sleeping mode, if the node resides within the range in which it may meet the ferry again. In addition, the node in the searching mode may switch to the sleeping mode, if it moves away from the range in which it may meet the ferry again.

Fig. 4 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.

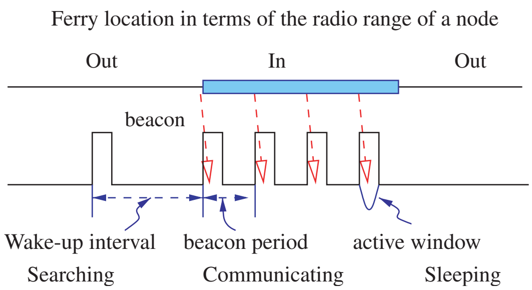

Fig. 4. Power management modes of a node depending on the ferry location.

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 doze or to turn off its radio 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 turns off its radio. Otherwise, it dozes. 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 3.2 and 3.3. 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 may result in missing beacons that 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 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 notations.

A ferry location is represented as $F(t)$ at time t. The route itself is defined as F where $F = \{F(t) | t \geqslant 0\}$. For a loosely scheduled scenario, $F(t)$ represents the estimated ferry location at t, assuming the ferry moves at its maximum speed without pause. 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.

3.2.1. 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. Fig. 5 shows how sleeping

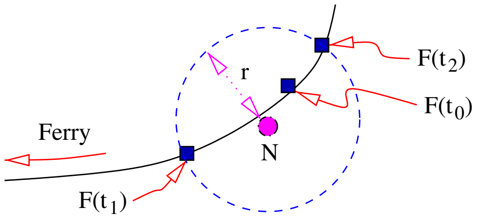

Fig. 5. The intersection of a ferry route and the radio range of a node.

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 needs 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.

3.2.2. 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.

3.3. 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.[5]

3.3.1. 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 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 Eq. (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.

Fig. 6 illustrates the above procedure. Currently, a node is located at $N(t_0)$. As time is incremented by 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

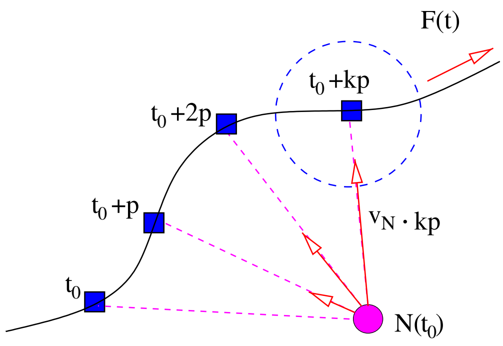

Fig. 6. Sleeping time estimation when a node movement is not known in advance.

[5] 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 $F(t) - N(t)$.

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.

3.3.2. 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 of 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 $\mathrm{d}(F, N(t_0)) = \min_t |F(t) - N(t_0)|$. 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

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

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

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

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:

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

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 Eq. (3).

Fig. 7 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

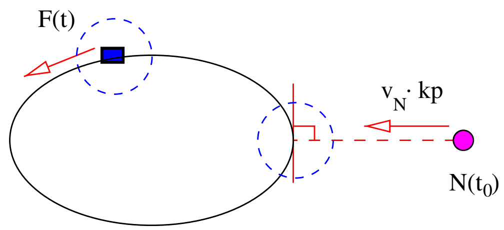

Fig. 7. Checking whether a node is in the feasible radio range of a ferry route.

and the line connecting the location and a node location intersects at 90°. 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 by varying wake-up intervals.

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. 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) [4]. We choose dynamic source routing (DSR) because it was determined to be the most efficient multihop routing protocol in [22].[6] In order to provide a fair comparison

[6] Other routing protocols, not compared in [22] but used in the design of power management mechanisms, tend to have specific movement or location restrictions. Thus, they may cause more overhead than DSR if used in general environments: For example, geographic forwarding requires frequent broadcasting, while not improving energy savings if the network density is low [9].

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 [8–13,15]. Assuming minimal spatial redundancy and no secondary low-power channel, synchronous and asynchronous wake-up mechanisms are the basic wake-up approaches to use.[7] 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 [15]. 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 four metrics:

- *Energy consumption per node*: An average energy consumption per node in the network.
- *Delivery delay*: An average delay per delivered message.
- *Delivery rate*: The ratio of successfully delivered messages to the total number of generated messages.
- *Energy cost*: 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 2000 m × 500 m region. We use 802.11 MAC and the default power setting of ns-2. For example, the radio range is 250 m and the data transmission rate is 2 Mb/s. Additional power usage parameters are shown in Table 1. The energy used for node mobility is not counted because we assume nodes have other means to cause their movement. For example, they could be devices carried by people or sensors attached to animals, where their movement occurs due to the physical movement of people or animals. In MF, the ferry uses the same setting as nodes. Also, the energy used for ferry mobility is not counted because we assume the mobility already exists for other purposes.

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 s. The sources send out messages at a constant rate of one message every 30 s for 3000 s. We define the traffic load as the total number of bytes generated from all sources in the entire simulation. Each message is of size 1 KB and has a timeout value of 5000 s after which messages not reaching their destinations 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 s 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 s and the active window is 500 ms. 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 100 ms for three wireless interfaces in [3]. To assign reasonably large energy consumption, we use 0.05636 J as the transition overhead to turn on the radio, which is equivalent to the amount of energy to transmit

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

data for 200 ms.[8] With this overhead, a node turns off its radio only if the expected sleep time is great enough to save energy beyond the overhead to turn on the radio back in both MF and DSR approaches. Otherwise, the node dozes and consumes no additional energy to enable the radio back.

To simulate node movement scenarios, we use the Random Way-point model [22] as follows. Each node selects a random destination in the region and moves toward the destination at a speed selected randomly between 0 and 10 m/s. When it reaches the destination, it pauses for a *pause time*, which is exponentially distributed with an average of 10 s. 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 10 m/s. In the scenario of loosely scheduled ferry movement, the ferry moves at a constant speed of 10 m/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 s.

4.2. Impact of traffic load

In this section, we evaluate the performance of MF and DSR under different traffic loads to show the relative robustness of the MF approach while the DSR approach suffers as more traffic load is injected to the network.

4.2.1. 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 s. In Fig. 8, 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 s. 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.

Fig. 8(a) shows the average energy consumption of nodes. Here, MF and DSR with large wake-up intervals (e.g., 50 or 200 s) significantly outperform DSR:CAM and DSR-2 under all traffic load. For example, in case of the traffic load of 6 MB, MF and DSR-200 consume only 0.2 J, while DSR-2 and DSR:CAM consume five times or nine times of that, respectively. This is expected because nodes sleep for longer time.

Fig. 8(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 Fig. 8(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 s 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 200–600 s, while that of DSR:CAM and DSR-2 is under 20 s for all traffic load.

Fig. 8(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 9 MB, 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.

[8] We simulated various overhead from the amount of energy for a node to transmit data for 50 ms to that for 1 s. As the overhead increases, the energy consumption increases. However, the amount of increase was too small to make any difference in our comparison.

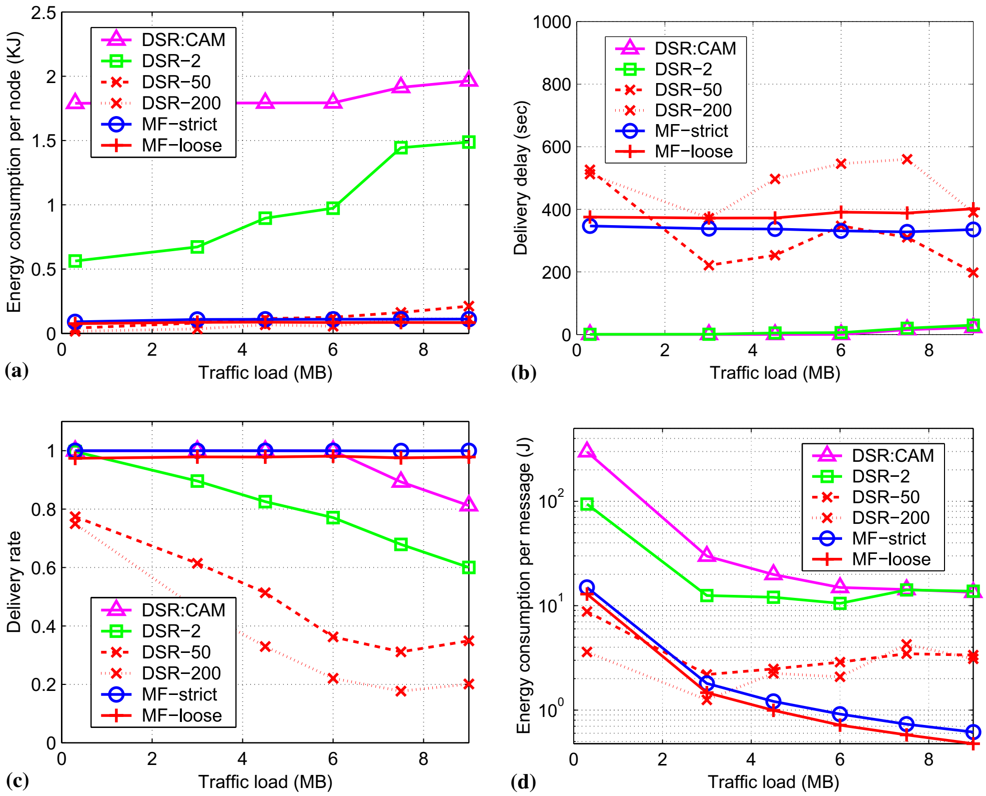

Fig. 8. Impact of traffic loads when nodes are stationary: (a) energy consumption (kJ/node), (b) delivery delay, (c) delivery rate, (d) energy cost (J/message).

In addition, Fig. 8(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 10 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 s, the messages are dropped. However, the loss rate is only 2%.

Fig. 8(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 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.

Finally, Fig. 8(d) shows that MF has less energy cost in a loose schedule scenario than in a strict schedule scenario because of less energy consumption. In the loose schedule scenario, a node spends more time in the searching mode and less time in the communicating mode out of the total simulation time than in the strict schedule scenario. Since a node wakes up 10 times more often in the communicating mode than in the searching mode, the total energy consumption of a node in the loose schedule scenario is less than that of a node in the strict schedule scenario. As a result, the energy cost may become smaller in the loose schedule scenario than in the strict schedule scenario.

4.2.2. Mobile nodes

In this section, we evaluate the performance of MF and DSR when nodes are mobile. The results shown in Fig. 9(a) and (d) are similar to those of the stationary node case. We, therefore, focus on the different features that show up in the simulation experiments.

Fig. 9(b) shows the delivery delay. Compared with the stationary node case, MF delivers faster when nodes are mobile because mobile nodes meet

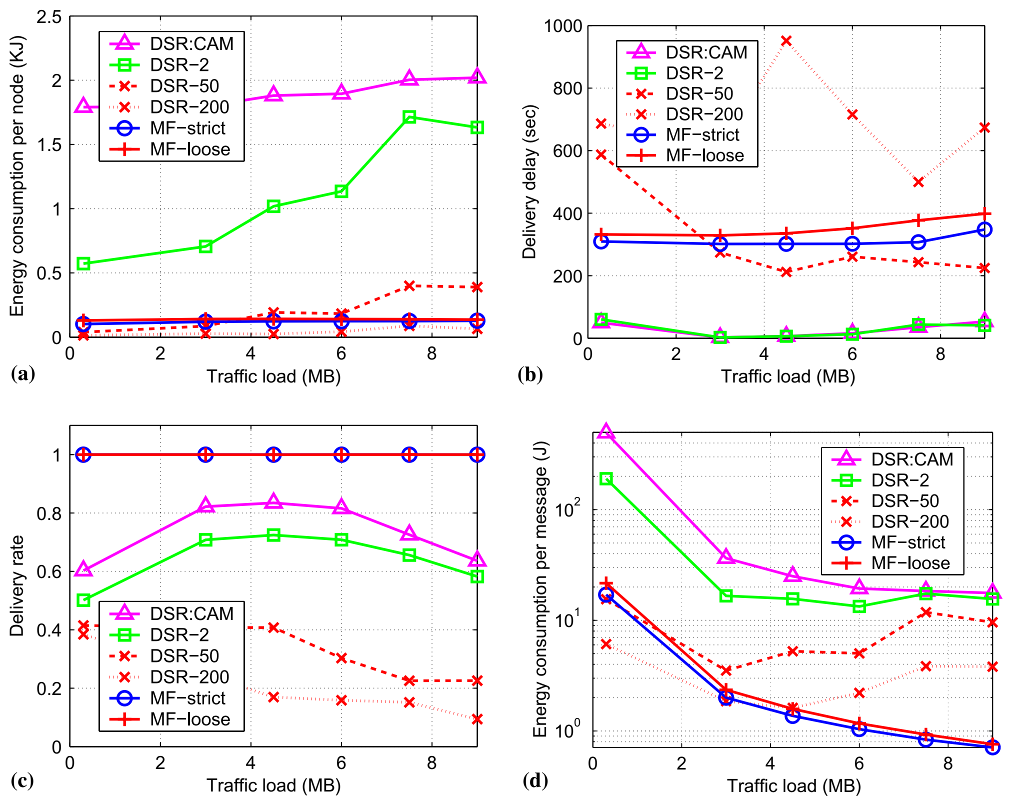

Fig. 9. Impact of traffic loads when nodes are mobile: (a) energy consumption (kJ/node), (b) delivery delay, (c) delivery rate, (d) energy cost (J/message).

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 Fig. 9(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 300 KB than when it is 3 MB. 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 3 MB, 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.

For the loose schedule scenario, the pause time of a ferry causes time shift between expected meeting time and real meeting time between the ferry and nodes. We simulated with various mean pause times from 0 to 420 s, which is the minimum time for a ferry to come back at the same location. However, the impact of pause time was too negligible to show as graphs. Therefore, we only summarize the results here. As the mean pause time increases, the ferry takes longer time to visit nodes. As a result, the delivery delay increases and the delivery rate decreases. Also, nodes spend more time in the searching mode instead of the sleeping mode because the real meeting time is much later than the expected meeting time, which increases energy consumption. The upper bound of this increase is the energy consumption that occurs when nodes are in the searching mode all the time except when they are in the communicating mode. This case occurs when nodes are mobile and the ferry has a loose schedule in our simulation. In such a scenario, our algorithm allows mobile nodes to sleep only when they become out of radio range from any point of the ferry route. Since the current ferry route covers the whole deployment area, a node would never become out of radio range from the ferry route and would never enter the sleeping mode. Thus, the energy consumption of this scenario provides the upper bound of energy consumption, and that is only slightly greater than that of DSR-50. In our simulation, the energy cost was also equivalent to that of DSR-50 when the traffic load is 3 MBps.

4.3. Impact of node mobility

In this section, we evaluate the performance of MF and DSR as node speeds vary from 5 to 50 m/s. This evaluation shows the robustness of the MF approach while the DSR approach suffers from network topology changes. Because the delivery rate of DSR-50 and DSR-200 is too low, we consider only DSR-2 and DSR:CAM.

Fig. 10(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.

Fig. 10(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 mobility force nodes to probe routing paths again. Waiting for the results of route probing adds up

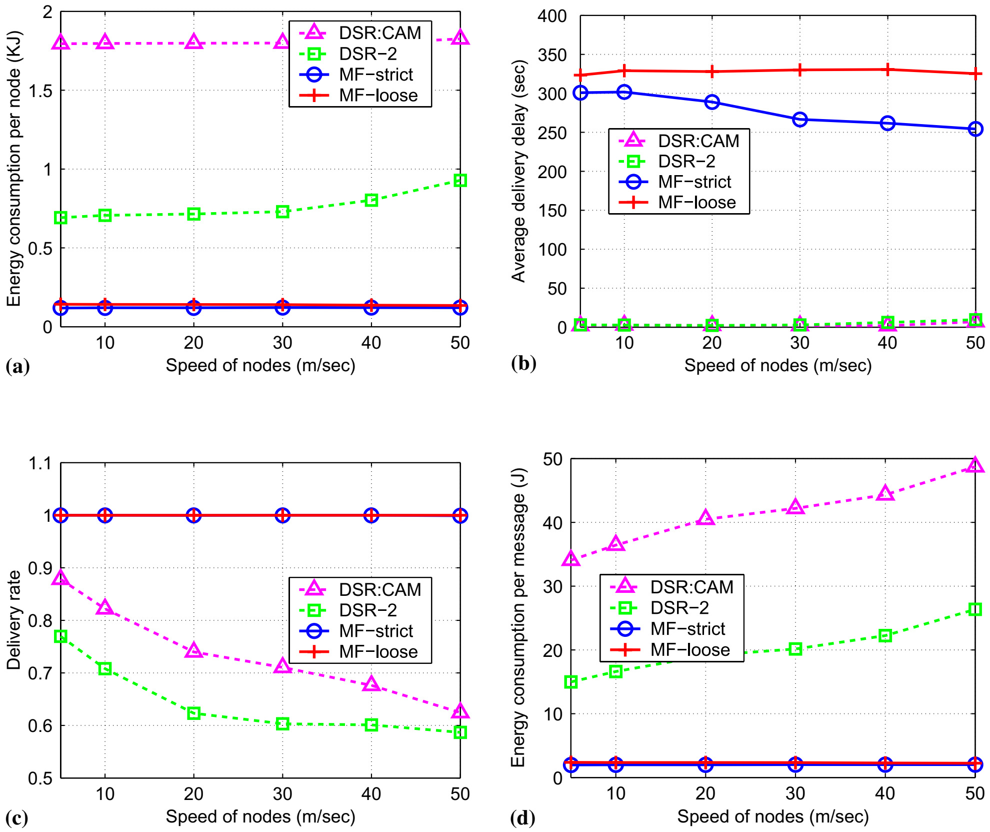

Fig. 10. Impact of node speeds: (a) energy consumption (kJ/node), (b) delivery delay, (c) delivery rate, (d) energy cost (J/message).

to the delivery delay. However, it is minor compared with the delivery delay of MF.

Fig. 10(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.

Fig. 10(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.

4.4. Impact of message timeout

In this section, we evaluate the performance of MF and DSR by varying the message timeout from 420 s to 4200 s. The timeout value of 420 s is a reasonable lower end for MF to be suitable because a ferry in our simulation setting usually takes more time to come back at the same location on its route. Because the message timeout enforces a maximum latency for each message delivery, this simulation provides the comparison between the MF and DSR approaches for a given fixed latency requirement.

Fig. 11(a) shows that DSR:CAM delivers more than 95% of messages for all timeout values. However, when the timeout value is as small as 420 s, the delivery rate of MF drops to 68% in the strict schedule case and to 37% in the loose schedule case because the ferry may not meet destination nodes within the timeout. At the same time, the delivery rate of DSR with power management also drops because sleeping nodes forces their neighbors to store messages until they wake up to receive messages, which increases the delivery delay of messages. Also, power management reduces the time for nodes to exchange messages among themselves, which causes more contention and retransmission. The resulting retransmission increases the delivery delay of messages, which causes message drop due to timeout. On the other hand, Fig. 11(b) shows that DSR:CAM has the highest energy cost. Therefore, DSR:CAM can deliver most of messages within given timeout values at the cost of energy.

4.5. 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 because the parameter trades between energy consumption and delivery delay. We vary the wake-up interval from 2 s to 200 s. We also vary the speed of the ferry as 5, 10, and 25 m/s. In Figs. 12 and 13, MF-x m/s represents the case of MF with power management where the speed of the ferry is x m/s, and its pause time at four corners of the route is exponentially distributed with an average of 50 s.

4.5.1. Stationary nodes

We first evaluate the performance of MF when nodes are stationary. Fig. 12(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.

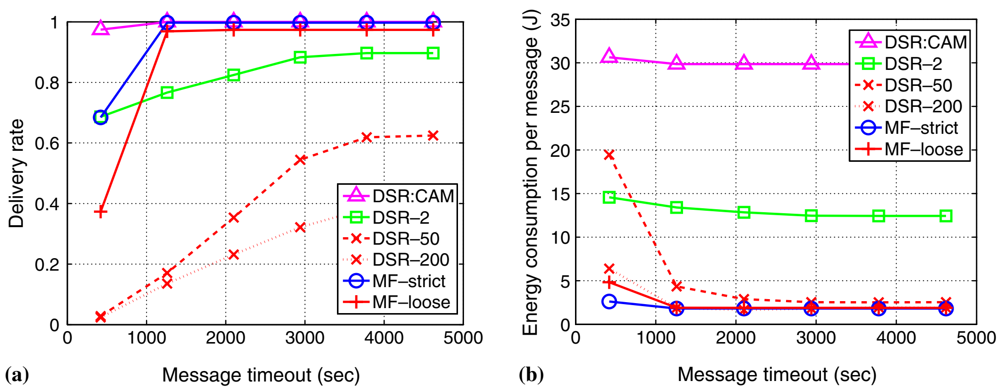

Fig. 11. Impact of message timeout when nodes are stationary: (a) delivery rate, (b) energy cost (J/message).

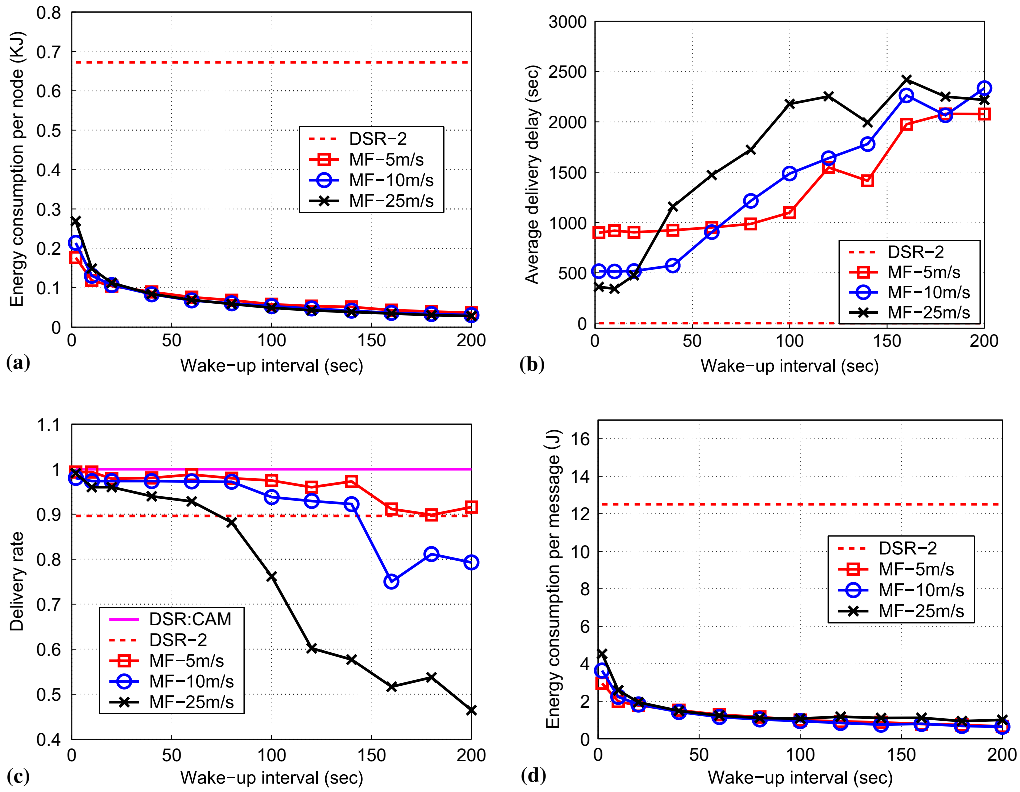

Fig. 12. Impact of wake-up intervals in the searching mode where nodes are stationary and the ferry moves on a loosely scheduled route: (a) energy consumption (kJ/node), (b) delivery delay, (c) delivery rate, (d) energy cost (J/message).

Fig. 12(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 when the ferry comes back. Fig. 12(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, when the wake-up interval is longer than 20 s, a faster ferry stays in the radio range of a node for a short period of time, which increases the probability for a node to miss the ferry. As a result, the delivery delay of a faster ferry is longer than that of slower ferries when the wake-up interval is long.

Fig. 12(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 s, they are dropped. Thus, a larger wake-up interval causes more message drops. Fig. 12(c) also shows that the delivery rate of a faster ferry is lower than that of a slower ferry because more messages are dropped due to timeout. Finally, Fig. 12(d) shows that the energy cost decreases as the wake-up interval increases due to overall energy savings.

4.5.2. Mobile nodes

We now evaluate the performance of MF when nodes are mobile. Fig. 13(a) and (b) show the trade-off between energy and delay similarly to Fig. 12. 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, Fig. 13(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 s. Because the radio range of the ferry swipes through the entire region over time, the ferry provides better network connection than DSR when nodes are mobile. Finally, Fig. 13(d) shows that the energy cost decreases as the wake-up interval increases.

In summary, both stationary and mobile nodes can save energy by increasing wake-up interval in

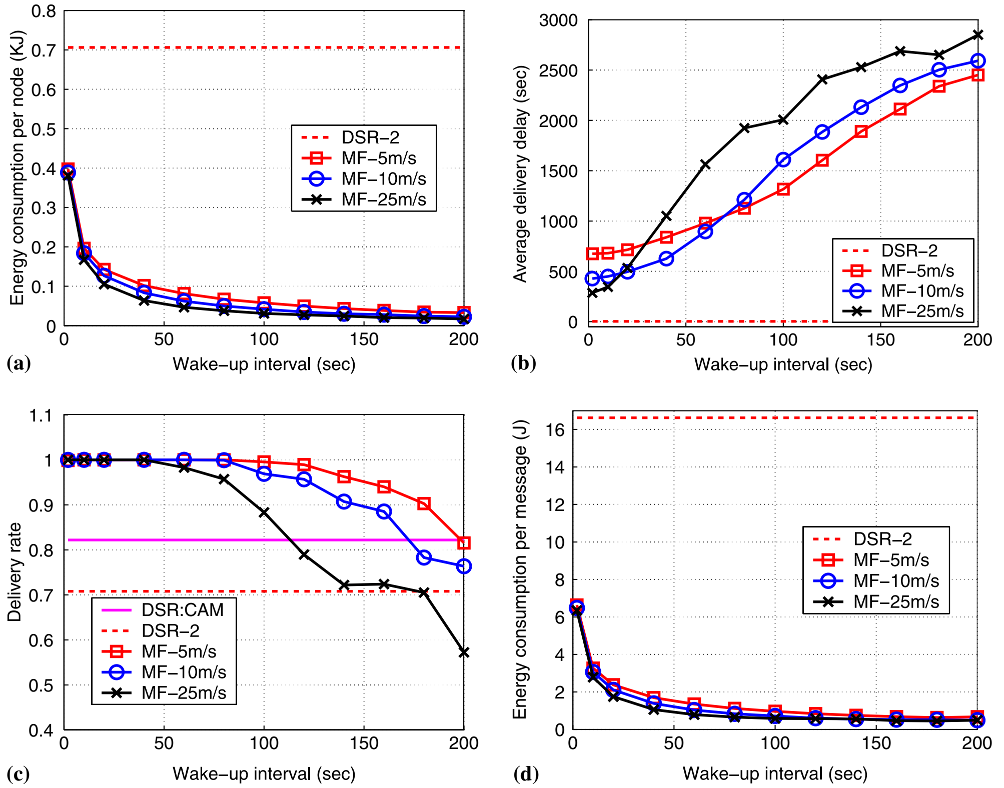

Fig. 13. Impact of wake-up intervals in the searching mode where nodes are mobile and the ferry moves on a loosely scheduled route: (a) energy consumption (kJ/node), (b) delivery delay, (c) delivery rate, (d) energy cost (J/message).

the searching mode. However, a very large wake-up interval reduces the delivery rate. Therefore, there is an optimal range of the wake-up interval in each network, and the wake-up interval that is less than 50 s is appropriate in these scenarios.

5. Related work

In this section, we review some related work on power management in wireless networks in five categories: cycling between sleep and waking up, designing energy efficient MAC protocols in dense networks, using mobility, tuning power management modes according to traffic patterns, and trading latency for energy.

Sleep/wake-up cycling mechanisms have been developed based on measurement studies [2,3], 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, many sleep/wake-up cycling mechanisms have been developed to save energy while keeping network connectivity. They fall into four categories: *synchronous*, *asynchronous*, *cell-based*, and *on-demand mechanisms*. In synchronous mechanisms such as 802.11 power saving mode (PSM) [8], 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 [12,15], neighboring nodes wake up in the way their awake time intervals overlap 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 [9,13], 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 [11,10], 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.

In a dense wireless network, transmission of multiple nodes interfere with each other, so at most two nodes can communicate within a radio range at a time. Thus, if a node sleeps while others are communicating, it can save energy without sacrificing its throughput. To achieve energy savings in this way, many medium access control (MAC) protocols have been proposed [23–26]. All these MAC protocols aim to increase sleeping time based on traffic activity in the neighborhood while keeping the network connectivity. Our MF approach delegates the responsibility of connecting a network to a ferry for more efficient energy savings. Also, these MAC protocols can be combined in our approach to optimize the energy savings in the communication mode, if multiple nodes tend to be in the radio range of a ferry at the same time.

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 [17–19,27,28]. Jain et al. [27] 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 did not 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. On the other hand, Goldenberg et al. propose to control node mobility to optimize the network performance such as energy efficiency based on specific traffic demands while using multihop routing protocols [29]. This approach requires nodes with extra capability to control their movement, while our approach simply utilizes the existing node mobility.

Traffic patterns are also important factors in the design of power management mechanisms. Zheng and Kravets [30] 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. [31] 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 to improve the energy savings when a ferry is in the radio range of a node.

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 [32,33,14]. The power management mechanisms in MF trade between latency and energy according to different network characteristics.

6. Conclusions

In this paper, we investigate the use of the message ferrying routing scheme in a densely deployed network 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 achieving high delivery rate. 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. Therefore, delay tolerant applications can use MF rather than a multihop routing protocol to save energy when both routing approaches are available.

References

[1] R.H. Kravets, Cooperative solutions to the dynamic management of communication resources, Ph.D. thesis, 1998.
[2] L.M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc network, in: INFOCOM, 2001.

[3] M. Stemm, R.H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, IEICE Trans. Commun. E80-B (8) (1997) 1125–1131.
[4] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: Mobile Computing, 1996.
[5] V.D. Park, M.S. Corson, Highly adaptive distributed routing algorithm for mobile wireless networks, in: INFOCOM, 1997.
[6] C.E. Perkins, E.M. Roye, Ad-hoc on-demand distance vector routing, in: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, 1999.
[7] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: SIGCOMM, 1994.
[8] Wireless LAN medium access control and physical layer specifications, IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee) (August 1999).
[9] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: MobiCom, 2001.
[10] E. Shih, P. Bahl, M.J. Sinclair, Wake on wireless: an event driven energy saving strategy for battery operated devices, in: MobiCom, 2002.
[11] C. Shurgers, V. Tsiatsis, S. Ganeriwai, M. Srivastava, Topology management for sensor networks: exploring latency and density, in: MobiHoc, 2002.
[12] Y.-C. Tseng, C.-S. Hsu, T.-Y. Hsieh, Power-saving protocols for IEEE 802-11-based multi-hop ad hoc networks, in: INFOCOM, 2002.
[13] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: MobiCom, 2001.
[14] Y. Yu, B. Krishnamachari, V. Prasanna, Energy-latency tradeoffs for data gathering in wireless sensor networks, in: INFOCOM, 2004.
[15] R. Zheng, R. Kravets, Asynchronous wakeup for ad hoc networks, in: MobiHoc, 2003.
[16] W. Zhao, M.H. Ammar, Message ferrying: proactive routing in highly partitioned wireless ad hoc networks, in: FTDCS, 2003.
[17] W. Zhao, M.H. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: MobiHoc, 2004.
[18] P. Juang, H. Oki, Y. Wang, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, in: ASPLOS-X, San Jose, CA, 2002.
[19] A.A. Hasson, R. Fletcher, A. Pentland, DakNet: a road to universal broadband connectivity, in: Wireless Internet UN ICT Conference Case Study, 2003, Available from: <http://www.medialabasia.org/>.
[20] S. Jain, R. Shah, W. Brunette, G. Borriello, 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.
[21] W. Zhao, M.H. Ammar, E. Zegura, Controlling the mobility of multiple data transport ferries in a delay tolerant network, in: INFOCOM, 2005.
[22] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Mobicom, 1998.
[23] S. Singh, C. Raghavendra, PAMAS: power aware multi-access protocol with signalling for ad hoc networks, ACM Comput. Commun. Rev. 28 (3) (1998) 5–26.
[24] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: INFOCOM, 2002.
[25] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: The Second ACM Conference on Embedded Networked Sensor Systems (SenSys), 2004.
[26] W.S. Conner, J. Chhabra, M. Yarvis, L. Krishnamurthy, Experimental evaluation of synchronization and topology control for in-building sensor network applications, in: The Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), 2003.
[27] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data MULEs: modeling a three-tier architecture for sparse sensor networks, in: IEEE Workshop on Sensor Network Protocols and Applications (SNPA), 2003.
[28] K. Fall, A delay-tolerant network architecture for challenged internets, in: SIGCOMM, 2003.
[29] D.K. Goldenberg, J. Lin, A.S. Morse, B.E. Rosen, Y.R. Yang, Towards mobility as a network control primitive, in: Mobihoc, 2004.
[30] R. Zheng, R. Kravets, On-demand power management for ad hoc network, in: INFOCOM, 2003.
[31] M. Anand, E.B. Nightingale, J. Flinn, Self-tuning wireless network power management, in: MobiCom, 2003.
[32] B. Prabhakar, E.U. Biyikoglu, A.E. Gamal, Energy-efficient transmission over a wireless link via lazy packet scheduling, in: INFOCOM, 2001.
[33] R.R. Kompella, A.C. Snoeren, Practical lazy scheduling in sensor networks, in: Sensys, 2003.

Hyewon Jun is a Ph.D. student in College of Computing at Georgia Institute of Technology, Atlanta, Georgia. She received her B.S. and M.S. degrees on mathematics from Yonsei University, Korea, in 1996 and 1998, respectively. Her research interests are in the design of energy efficient wireless ad hoc and sensor networks, mobility management, multimedia service architecture, distributed network architecture, protocols, and applications.

Wenrui Zhao received his B.S. degree from University of Science and Technology of China in 1996 and his M.S. degree from Chinese Academy of Sciences in 1999, both in computer science. He is currently a Ph.D. candidate at College of Computing, Georgia Institute of Technology. His research interests are generally in computer networking, including wireless mesh networks, mobile ad hoc networks, sensor/actuator networks, and delay/disruption tolerant networks. He is a student member of ACM and IEEE.

Mostafa Ammar is a Regents' Professor with the College of Computing at Georgia Tech. He has been with Georgia Tech since 1985. He received the S.B. and S.M. degrees from the Massachusetts Institute of Technology in 1978 and 1980, respectively and the Ph.D. in Electrical Engineering from the University of Waterloo, Ontario, Canada in 1985. His research interests are in network architectures, protocols and services. He has contributions in the areas of multicast communication and services, multimedia streaming, content distribution networks, network simulation and most recently in disruption-tolerant networks. He was the co-recipient of the Best Paper Awards at the 7th WWW conference for the paper on the "Interactive Multimedia Jukebox" and the 2002 Parallel and Distributed Simulation (PADS) conference for the paper on "Updateable Network Simulation". He served as the Editor-in-Chief of the IEEE/ACM Transactions on Networking from 1999 to 2003. He is a Fellow of the IEEE and a Fellow of the ACM.

Ellen W. Zegura received the B.S. degree in Computer Science (1987), the B.S. degree in Electrical Engineering (1987), the M.S. degree in Computer Science (1990) and the D.Sc. in Computer Science (1993) all from Washington University, St. Louis, Missouri. Since 1993, she has been on the faculty in the College of Computing at Georgia Tech. She received tenure and promotion to Associate Professor in 1999. She was promoted to Full Professor in 2004. She is currently Associate Dean and Chair of the Computing Science and Systems Division. Her research area is computer networking, with interests in content distribution, wide-area (Internet) topology and routing, and mobile wireless networking. She served as Editor-in-Chief of IEEE/ACM Transactions on Networking from 2003 to 2005, and she has served on the program committees of numerous networking conferences.

Chungki Lee is a professor in the Department of Computer Software at Myongji University, Korea. He received the B.S. degree from the Seoul National University, Seoul, Korea in 1979, and the M.S. and Ph.D. in Computer Science from the Georgia Institute of Technology, Atlanta, USA in 1990 and 1993, respectively. For the years 1994–1995 he worked at the National Computerization Agency in Korea as a Senior Member of Technical Staff. His research interests are in the areas of mobile network protocols, mobile computing, and computer education.