A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks *

Wenrui Zhao, Mostafa Ammar and Ellen Zegura
College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332
{wrzhao, ammar, ewz}@cc.gatech.edu

ABSTRACT

Mobile Ad Hoc Networks (MANETs) provide rapidly deployable and self-configuring network capacity required in many critical applications, e.g., battlefields, disaster relief and wide area sensing. In this paper we study the problem of efficient data delivery in *sparse* MANETs where network partitions can last for a significant period. Previous approaches rely on the use of either long range communication which leads to rapid draining of nodes' limited batteries, or existing node mobility which results in low data delivery rates and large delays. In this paper, we describe a *Message Ferrying* (MF) approach to address the problem. MF is a mobility-assisted approach which utilizes a set of special mobile nodes called *message ferries* (or *ferries* for short) to provide communication service for nodes in the deployment area. The main idea behind the MF approach is to introduce *non-randomness* in the movement of nodes and exploit such non-randomness to help deliver data. We study two variations of MF, depending on whether ferries or nodes initiate proactive movement. The MF design exploits mobility to improve data delivery performance and reduce energy consumption in nodes. We evaluate the performance of MF via extensive *ns* simulations which confirm the MF approach is efficient in both data delivery and energy consumption under a variety of network conditions.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; C.2.2 [**Network Protocol**]: Routing protocols

General Terms

Algorithms, Design, Performance

Keywords

Mobile ad hoc networks, mobility-assisted data delivery, sparse networks

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

MobiHoc'04, May 24–26, 2004, Roppongi, Japan.

1. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are multi-hop networks in which wireless mobile nodes cooperate to maintain network connectivity and perform routing functions [17, 25]. MANETs enable nodes to communicate with each other without any existing infrastructure or centralized administration. These rapidly deployable and self-configuring networks have applications in many critical areas, such as battlefields, disaster relief and wide area sensing and surveillance.

This paper studies the problem of efficient data delivery in *sparse* mobile ad hoc networks. Specifically, we focus on mobile networks where nodes are sparsely distributed such that network partitions can last for a significant period. Sparse networks naturally arise in a variety of applications. For example, imagine the following hypothetical disaster scenario. A severe earthquake has occurred which collapses buildings, traps people in the debris, damages utilities and roads, and causes fires and explosions. Under this situation, the ability to communicate, even at low rates, is extremely valuable for sharing vital information (such as the number and locations of survivors, damages and potential hazards) and coordinating rescue efforts. However, providing communication capacity is difficult. First, fixed and stable communication infrastructure might be destroyed. Even if some infrastructure is usable, most rescue participants and victims may not have access to it. Second, available devices such as cell phones or PDAs can only communicate within a limited range. Due to the size of the area affected, a connected ad hoc network can not be formed using these devices alone.

Routing in ad hoc networks has been an active research field in recent years, producing many routing algorithms such as DSR, DSDV and AODV [17, 25, 26]. However, most of the existing work focuses on *connected* networks where an end-to-end path exists between any two nodes in the network. In sparse networks, where partitions are not exceptional events, these routing algorithms will fail to deliver packets because no route is found to reach their destinations. To overcome partitions in sparse networks, a straightforward approach is to use radios with longer transmission ranges and maintain persistent network connectivity. However, since many mobile nodes use batteries for power supply, the use of a long range radio leads to excessive energy consumption. In addition, the availability of such devices in critical scenarios would be questionable.

Previous work proposes mobility-based approaches which use short range communication and exploit node mobility to help deliver data [33, 9, 21, 4, 29]. Specifically, nodes buffer and carry packets during network partitions, and forward packets to other nodes when they meet. This *store-carry-forward* paradigm is suitable for delay tolerant applications such as sensor data collection, messaging and file transfer. In general, these approaches can be classified as reactive schemes or proactive schemes. In *reactive* schemes such as Epidemic routing [33], applications rely on movement that is inherent in the devices themselves to help deliver messages. When disconnected, nodes passively wait for their own mobility to allow them to re-connect. Since encounters between nodes can be unpredictable and rare, these approaches suffer potentially low data delivery rates and large delays. To increase delivery rate and reduce delay, nodes typically propagate messages throughout the network, which, however, exacerbates contention for limited buffers in nodes and drains nodes' limited energy. In *proactive* approaches, nodes modify their trajectories proactively for communication purposes. Li and Rus [21] propose an optimal algorithm to compute the trajectories of nodes for minimizing message transmission delay. However, as pointed out by the authors, it is difficult to extend this algorithm to efficiently support multiple simultaneous transmissions.

In this paper, we describe a *Message Ferrying (MF)* approach for data delivery in sparse networks. MF is a proactive mobility-assisted approach which utilizes a set of special mobile nodes called *message ferries* (or *ferries* for short) to provide communication services for nodes in the network. Similar to their real life analog, message ferries move around the deployment area and take responsibility for carrying data between nodes. The main idea behind the Message Ferrying approach is to introduce *non-randomness* in the movement of nodes and exploit such non-randomness to help deliver data. Message ferrying can be used effectively in a variety of applications including battlefields, disaster relief, wide area sensing, non-interactive Internet access and anonymous communication. For example, in the earthquake disaster scenario, unmanned aerial vehicles or ground vehicles that are equipped with large storage and short range radios can be used as message ferries to gather and carry data among disconnected areas. This enables rescue participants and victims to use available devices such as cell phones, PDAs or smart tags for communication.

In a previous paper [35], we introduced the idea of Message Ferrying and studied its use in networks with *stationary* nodes. In this paper, we consider networks with *mobile* nodes. We develop two variations of the MF schemes, depending on whether ferries or nodes initiate non-random proactive movement. In the *Node-Initiated MF (NIMF)* scheme, ferries move around the deployed area according to known routes and communicate with other nodes they meet. With knowledge of ferry routes, nodes periodically move close to a ferry and communicate with the ferry. In the *Ferry-Initiated MF (FIMF)* scheme, ferries move proactively to meet nodes. When a node wants to send packets to other nodes or receive packets, it generates a service request and transmits it to a chosen ferry using a long range radio[1]. Upon reception of a service request, the ferry will adjust its trajectory to meet up with the node and exchange packets using short range radios. In both schemes, nodes can communicate with distant nodes that are out of range by using ferries as relays.

The Message Ferrying design is distinguished from other mobility-assisted approaches by its explicit exploitation of non-random node mobility and the use of message ferries, which improves data delivery and energy efficiency. In MF, most communication involves short range radios. Long range radios are only used in FIMF for small control messages, avoiding excessive energy consumption. By using ferries as relays, routing is efficient without the energy cost and the network load burden involved in other mobility-assisted schemes that use flooding. Our simulation results confirm the effectiveness and efficiency of the MF schemes.

[1]The FIMF scheme can be adapted if nodes do not have long range radios, however the present exposition assumes that ferries and nodes are so equipped.

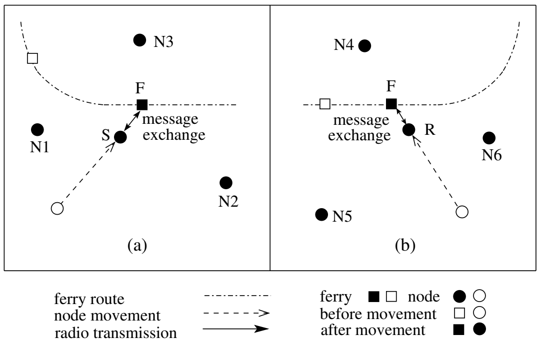


Figure 1: An example of message delivery in the node-initiated MF scheme.

The rest of this paper is structured as follows. Section 2 gives an overview of the MF approach and some potential applications, and describes the specific MF system considered in this paper. The node-initiated and ferry-initiated MF schemes are described in Section 3 and Section 4 respectively. Simulation results are presented in Section 5 to evaluate the effectiveness of the MF schemes. We discuss some related design issues and our future work in Section 6. Related work is reviewed in Section 7 and the paper is concluded in Section 8.

2. OVERVIEW OF MESSAGE FERRYING

The Message Ferrying (MF) scheme is a proactive approach for data delivery in sparse networks. It introduces *non-randomness* to node mobility and exploits such non-randomness to provide physical connectivity among nodes. In an MF scheme, the network devices are classified as *message ferries* (or *ferries* for short) or *regular nodes* based on their roles in communication. Ferries are devices which take responsibility of carrying messages among other nodes, while regular nodes[2] are devices without such responsibility. There are many different ways to introduce non-randomness in node movement. For example, in the node-initiated MF scheme described in Section 3, ferries move around the deployed area according to known routes, collect messages from regular nodes and deliver messages to their destinations or other ferries. With knowledge about ferry routes, nodes can adapt their trajectories to meet the ferries and transmit or receive messages. By using ferries as relays, nodes can communicate with distant nodes that are out of range (see Fig. 1 for an example).

Message ferrying is a broad concept, with the potential for many variations in specific design and implementation. We now explore a few possible contexts in which message ferrying can be used effectively.

2.1 Message Ferrying Applications

We envision that Message ferrying can be used effectively in the following four categories of applications.

Crisis-driven — This category includes battlefield and disaster applications, where fixed and stable infrastructure is limited or unavailable due to environmental conditions. For example, in a disaster relief effort, nodes equipped with short range radios may move out of range of one another. In a battlefield, equipment intended to provide connectivity may be compromised so that it becomes inoperable. In either scenario, the wide physical range of the deployed

[2]We will use the term "node(s)" to refer to regular nodes or to both regular nodes and ferries. The usage should be clear from the context.

area may prevent end-to-end connectivity. For these settings, message ferries enable communication that would otherwise be impossible. The limitations of message ferrying, such as low throughput and large delay as compared to connected networks, are acceptable because the alternative is no communication.

Geography-driven — This category includes wide area sensing and surveillance applications. While sensor networks are normally densely deployed, there are situations where sensor networks are inherently sparse due to the geographic span involved. For example, in the ZebraNet project [18], sensors are attached to zebras and used to study the behavior of wildlife. As zebras move, these sensors become spread throughout the area and form a sparse network. Other examples include DataMULE [29], SWIM [31] and the smart tag system in [4]. In these settings, applications can tolerate significant transfer delay which makes message ferrying a suitable solution.

Cost-driven — This category includes applications that could use other existing technologies, but where message ferries offer a cost-effective alternative. For example, in the DakNet project [1], vehicles such as buses are used to transport data between remote areas such as villages and cities to provide store-and-forward Internet access. A metropolitan government could include message ferries on public buses, providing low cost metro-area messaging.

Service-driven — This category includes applications that require a service not provided by other available networking infrastructure. For example, message ferrying could offer a privacy or anonymity service for message delivery that is not otherwise available, or may be available but not trusted (e.g., in a hostile political environment). Message ferrying allows by-passing the existing infrastructure to obtain a different service, though with degraded performance.

2.2 MF System under Consideration

As discussed above, there are several possible scenarios in which message ferrying can be used. Each scenario implies a particular set of functions and capabilities for the components (ferries and regular nodes). In this paper, we focus on the application of MF in sparse mobile networks. Specifically, regular nodes are assumed to have assigned tasks that involve movement in the deployment area and limited in resources such as battery, memory and computation power. Ferries are special mobile nodes which take responsibility for carrying data between regular nodes and have fewer constraints in resources, e.g., equipped with renewable power, large memory and powerful processors. The purposes of ferries are to provide communication capacity between regular nodes. For example, in a college campus, buses equipped with hard disks and wireless interfaces can act as ferries to provide messaging service to students; in battlefield and disaster relief environments, aerial or ground vehicles can be used as ferries to gather and carry data among disconnected areas.

Data transmission between regular nodes is in application layer data units called *messages*. Since nodes have limited memory, messages will be dropped when buffers overflow. In addition, each message carries with it a timeout value which specifies when this message should be dropped if not delivered. The setting of the timeout value reflects the delay requirement of applications. Message Ferrying is suitable for applications which can tolerate significant transfer delay, such as messaging, file transfer, email, data collection in sensor networks and other non-real-time applications. These applications would benefit from the eventual delivery of data even if the delay is moderate.

The design of the MF schemes is based on location-awareness and mobility. Each node or ferry is aware of its own location, for example through receiving GPS signals or other localization mechanism. The mobility of nodes and ferries can come in two varieties, which may co-exist in the same scenario:

- Task-oriented mobility: The ferry or node mobility is determined for non-messaging reasons. For example, the route of a campus bus acting as a message ferry is determined by passenger-carrying concerns; a PDA is carried around by a student moving inside a campus.
- Messaging-oriented mobility: The ferry or node mobility is specifically designed for improving the performance of messaging. For example, ferries are implemented in a subset of robots dispersed in a disaster area, and the mobility of the ferry robots is specifically optimized for maximizing the efficiency of messaging among the other robots.

In this paper, we focus on the case where a single ferry is used and there are no buffer or energy constraint in the ferry. This is the case, for example, when a shuttle bus (an airplane) acts as the ferry to transport data in a campus (battlefield). In addition, regular nodes are assumed to operate independently. We will discuss more general MF systems in Section 6. In the following sections, we will describe in detail two Message Ferrying schemes which utilize messaging-oriented mobility of either the ferry or regular nodes.

3. NODE-INITIATED MESSAGE FERRYING SCHEME

In the Node-Initiated MF (NIMF) scheme, the ferry moves according to a specific route. The ferry route is known by nodes, e.g., periodically broadcast by the ferry or conveyed by other out-of-band means. Nodes take proactive movement periodically to meet up with the ferry. Fig. 1 shows an example of how NIMF operates. In Fig. 1(a), the ferry *F* moves on a known route, part of which is illustrated. As the sending node *S* approaches the ferry, it forwards its messages to the ferry which will be responsible for delivery. In Fig. 1(b), the receiving node *R* meets the ferry and receives its messages. By using the ferry as a relay, *S* can send messages to *R* even there is no end-to-end path between them.

In the following, we will describe the operations of NIMF and how nodes adjust their movement to meet the ferry.

3.1 NIMF Operations

Fig. 2 shows a sketch of the node operations in the NIMF scheme. A node operates in 4 modes: `WORKING`, `GO_TO_FERRY`, `SEND/RECV`, and `GO_TO_WORK` (see Fig. 3 for the transitions among modes). A node is initially in the `WORKING` mode and moves according to its assigned task. The *trajectory control* mechanism of the node determines when it should proactively move to meet the ferry for sending or receiving messages. We describe this mechanism in detail in Section 3.2. The node enters the `GO_TO_FERRY` mode when it decides to go to the ferry, and approaches the ferry. When the node detects the ferry is within its transmission range, the node enters the `SEND/RECV` mode and exchanges messages with the ferry. After completing message exchange or the ferry has moved out of range, the node enters the `GO_TO_WORK` mode to return to its location prior to the detour. Upon return to the prior location, the node enters the `WORKING` mode. In addition, nodes can switch to the `SEND/RECV` mode from the `WORKING` mode when they meet the ferry "unintentionally", e.g., without proactive movement. In this case, the node returns to the `WORKING` mode after interacting with the ferry.

Fig. 4 presents the ferry operations. The ferry moves on a specified route and exchanges messages with nodes when they meet. To support this message exchange, the ferry and nodes must be able to

detour: whether the node is detouring;
mode: which mode the node is in;
1. `WORKING` mode
    detour = FALSE;
    IF *Trajectory Control* indicates time to go to the ferry,
        detour = TRUE;
        mode = `GO_TO_FERRY`;
    On reception of a `Hello` message from the ferry:
        mode = `SEND/RECV`;
2. `GO_TO_FERRY` mode
    Calculate a shortest path to meet the ferry;
    Move toward the ferry;
    On reception of a `Hello` message from the ferry:
        mode = `SEND/RECV`;
3. `SEND/RECV` mode
    Exchange messages with the ferry;
    On finish of message exchange or the ferry is out of range:
        IF *detour* is TRUE,
            mode = `GO_TO_WORK`;
        ELSE
            mode = `WORKING`;
4. `GO_TO_WORK` mode
    Move back to node's location prior to the detour;
    On return to the prior location:
        mode = `WORKING`;
    On reception of a `Hello` message from the ferry:
        mode = `SEND/RECV`;

Figure 2: Node operations in NIMF.

detect one another when they are close, e.g., by periodically broadcasting `Hello` messages. In MF, the use of the ferry releases nodes from the responsibility of transmitting `Hello` messages, thus saving node energy. Specifically, the ferry sends out `Hello` messages periodically using a short range radio, and nodes simply listen to the channel to detect the ferry. When a node receives a `Hello` message from the ferry, it will reply with an `Echo` message. After identifying each other, the node and the ferry initiate a message exchange conversation. The node will transmit all its buffered messages to the ferry which will be responsible for delivery. The ferry will then deliver to the node all messages buffered at the ferry and destined to the node.

Message forwarding in MF is simple: messages are forwarded from the source to the ferry, and then from the ferry to the destination[3]. This deterministic routing improves both data delivery and energy efficiency. In sparse networks, network partitions may last for significantly long periods and lead to buffer contention in nodes because messages can not be removed from buffers and new messages might be generated. In addition, flooding of messages such as in Epidemic routing [33] generates a large number of redundant messages which not only intensifies buffer contention, but also wastes energy. In MF, the use of the ferry as a relay avoids such buffer contention and redundant transmission problems.

3.2 Node Trajectory Control

The proactive movement of nodes to meet the ferry will generally degrade performance on the tasks that are assigned to nodes, because the node must detour from its intended path. Thus nodes need to strike a balance between performance gain in data delivery and performance degradation in assigned tasks resulting from such proactive movement. In applications where MF is useful, delivery rate is an important metric. Thus the goal of *trajectory control* is to minimize message drops while reducing the negative impact of proactive movement.

We propose a method for trajectory control that considers both impact on assigned task and message drop rate. We first consider message drops. Messages may be dropped in nodes because of message timeout or buffer overflow, and in the buffer-unlimited ferry because of timeout. In Section 3.3, we will describe how nodes and the ferry maintain information about message generation and drops. In this section, we assume this information is available in nodes and describe how it is used for trajectory control. Specifically, we consider a discrete time model, i.e., time is divided into fixed-length slots. Node i maintains $D_i^n(t)$ which is node i's own message drop rate during time slot t, and $D_i^f(t)$ which is the drop rate in the ferry for destination i during slot t. Let t_d be the time slot in which the node is expected to meet the ferry after proactive movement. t_d can be calculated based on knowledge of the ferry route, ferry speed and the node's location. If the node chooses not to meet the ferry at this time, it will incur message drops at a rate $D_i(t_d) = D_i^n(t_d) + D_i^f(t_d)$ during slot t_d. It may also incur future message timeouts. Let α be the average time between a node's visits to the ferry and T be the message timeout value in slots. Messages currently buffered in node i will be expected to reach their destinations during slot $t_d + \alpha$. Here instead of using $\alpha/2$, the expected time a message stays at the ferry before delivered to the destination, we use α to be conservative in message drop estimation. We also assume t_d is the same in the future because t_d depends on the future movement of the node which is unknown. So, if the node chooses not to move to the ferry, it will incur message timeouts at a rate $m_i(t_d + \alpha - T)$ during slot $t_d + \alpha$, where $m_i(t)$ is the message generation rate during slot t. Considering both kinds of message drops, we adopt a policy which allows a node to move to the ferry only when

$$(D_i(t_d) + m_i(t_d + \alpha - T))/(G_i^n + G_i^f) \geq \beta \qquad (1)$$

is true, where β is a predefined parameter, G_i^n is the message generation rate in node i, and G_i^f is the message arrival rate in the ferry for destination i. We will describe how G_i^n and G_i^f are obtained in Section 3.3. By using β, this policy tries to avoid node detour in situations where the message drop rates are low compared to the message generation rate.

We now turn to the issue of limiting the negative impact of proactive movement. The impact of proactive movement is generally application specific and may be unknown in advance. In this paper, we use a simplified model based on *work time percentage* (WTP) to represent the impact. WTP is defined as the percentage of time a node is free to work on assigned tasks (i.e., the node is not detouring for message transmission or reception). To meet the needs of assigned tasks, we specify that a node is allowed to proactively move to the ferry only when its WTP w_i is above a predefined threshold ω.

Putting together the above factors, we determine a policy that states that a node i modifies its trajectory only when $w_i > \omega$ and Eq. (1) is true. Once a node decides to go to the ferry, it will take a shortest trajectory to meet the ferry.

3.3 Message Drops

As described in Section 3.2, nodes determine when to meet the ferry using information about message generation and drops. In this section, we describe how this information is obtained. Basically, nodes and the ferry keep a history of previous message generation rates and based on that, compute the expected message drop rate and generation rate.

[3]When nodes communicate with the ferry via gateway nodes in the cluster, messages are forwarded from the source to the destination via a gateway nodes in the source cluster, the ferry and a gateway node in the destination cluster.

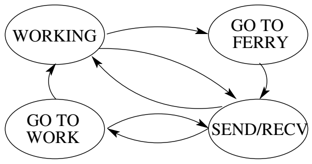


Figure 3: Mode transition diagram for nodes in NIMF.

1. Move according to a ferry route;
2. Broadcast `Hello` messages periodically;
3. On reception of an `Echo` message from a node:
   Exchange messages with the node;

Figure 4: Ferry operations in NIMF.

We first consider message drops in a node, say node i. Messages may be dropped for either buffer overflow or message timeout, when buffered at a node waiting for interaction with the ferry. For simplicity of exposition, we assume all messages have the same size and timeout value T. Suppose the length of a time slot is Δ. Let $m_i(t)$ be the message generation rate during slot t. Let $M_i(t)$ be the cumulative amount of messages generated over slots $[1,t]$, or $M_i(t) = \sum_{k=1}^{t} m_i(k)\Delta$. Assuming no buffer overflow, the message timeout rate during slot t is $a(t) = m_i(t-T)$ and messages being dropped are generated during slot $t_a = t - T$. Similarly, assuming no message timeout, the message drop rate during slot t because of buffer overflow is $b(t) = m_i(t)$, and messages being dropped are generated during slot $t_b = \max\{k : \sum_{j=k}^{t} m_i(j)\Delta > B_i\}$ where B_i is the node buffer size. In the description above, we do not consider the cases when $t < T$ or $M_i(t) \leq B_i$ for simplicity of exposition, which can be easily handled in our model. Now we compute the message drop rate $D_i^n(t)$ for node i during slot t as follows.

$$D_i^n(t) = \begin{cases} a(t) & \text{if } t_a > t_b, \\ \max\{a(t), b(t)\} & \text{if } t_a = t_b, \\ b(t) & \text{if } t_a < t_b. \end{cases} \tag{2}$$

For example, when $t_a < t_b$, buffer overflow occurs before the oldest message times out. So messages are dropped because of buffer overflow and the drop rate is $b(t)$. Fig. 5 illustrates an example of the computation of message drop rate.

Message drops may also occur in the ferry. Because we assume the ferry buffer is not a limitation, these drops occur only because messages timeout before the destination node approaches the ferry. The ferry maintains message drop information and conveys this information to nodes. Specifically, for each destination node i, the ferry maintains the message generation rate $m_i^f(t)$ and the cumulative amount of messages $M_i^f(t)$. The ferry then computes the message drop rate $D_i^f(t)$ for each node i. Periodically the ferry broadcasts a `Ferry_Status` message to all nodes using a long range radio and including $D_i^f(t)$ for all i. To conserve bandwidth, the ferry may only transmit an approximate version of $D_i^f(t)$, e.g., including a subset of all $(t, D_i^f(t))$ values in the messages.

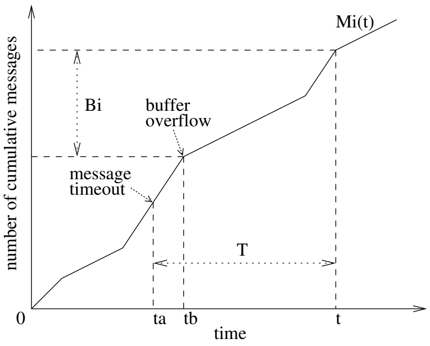


Figure 5: An example of message drop rate computation in node i. Messages dropped during slot t would have arrived at node i during either slot t_a or slot t_b (t_b in this example).

Nodes also estimate their message generation rates G_i^n, e.g., using a sliding window that averages over past history. G_i^n is used to compute the message drop rate $D_i^n(t)$ for future time t. Similarly, the ferry maintains the message generation rate G_i^f for each destination node i and broadcasts this information to nodes via `Ferry_Status` messages.

4. FERRY-INITIATED MESSAGE FERRYING SCHEME

In the Ferry-Initiated Message Ferrying (FIMF) scheme, the ferry takes proactive movement to meet up with nodes for communication purposes. We assume that the ferry moves faster than nodes. In addition, we assume that nodes are equipped with a long range radio which is used for transmitting control messages. Note that while the ferry can broadcast data to all nodes in the area, the transmission range of nodes' long range radios may not necessarily cover the whole deployment area due to power constraints.

Fig. 6 shows a simplified example of how the FIMF scheme operates. Initially the ferry F follows a specific *default route* and periodically broadcasts its location to nodes using a long range radio. When a node S finds the ferry is nearby and wants to send or receive messages via the ferry, it sends a `Service_Request` message to the ferry using its long range radio (Fig. 6(a)). This message contains the node's location information. Upon reception of a request message, the ferry adjusts its trajectory to meet the node. To guide the ferry movement, the node occasionally transmits `Location_Update` messages to notify the ferry of its new location (Fig. 6(b)). When the ferry and the node are close enough, they exchange messages via short range radios (Fig. 6(c)). After completing message exchange with the node, the ferry moves back to its default route (Fig. 6(d)).

4.1 FIMF Operations

Fig. 7 shows a sketch of the operations of nodes. A node can be in two modes: `DISASSOCIATED` and `ASSOCIATED`. A node is initially in the `DISASSOCIATED` mode, meaning that it has not requested service from the ferry. The *notification control* mechanism, discussed in Section 4.2, determines whether the node should send a `Service_Request` message to the ferry. After sending a request message to the ferry, the node enters the `ASSOCIATED` mode and waits for the interaction with the ferry. When a node is in the `ASSOCIATED` mode, notification control determines when to send a `Location_Update` message to notify the ferry of the node's new location. In both modes, the node may exchange messages with the ferry if it is close to the ferry and receives a `Hello` message from the ferry. After interaction with the ferry, the node returns to the `DISASSOCIATED` mode.

Fig. 8 shows the ferry operations. The ferry operates in two modes: `IDLE` and `WORKING`. Initially the ferry is in the `IDLE` mode and follows a specific *default route*. It periodically broadcasts its location information to nodes via a long range radio. Upon the reception of a `Service_Request` message from a node, the ferry switches to the `WORKING` mode. In the `WORKING` mode, the

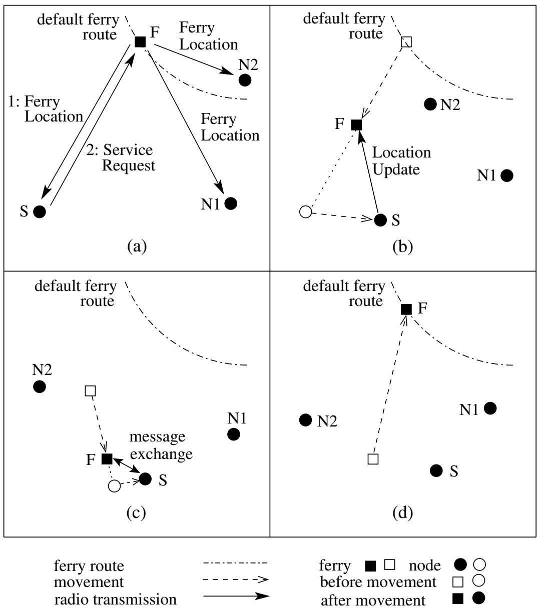


Figure 6: An example of FIMF operations.

NC: notification control scheme
mode: which mode the node is in;
1. `DISASSOCIATED` mode
    IF *NC* indicates sending a `Service_Request` message,
        Send a `Service_Request` message to the ferry;
        mode = `ASSOCIATED`;
    On reception of a `Hello` message from the ferry:
        Exchange messages with the ferry;
2. `ASSOCIATED` mode
    IF *NC* indicates sending a `Location_Update` message,
        Send a `Location_Update` message to the ferry;
    On reception of a `Hello` message from the ferry:
        Exchange messages with the ferry;
        On finish of interacting with the ferry;
            mode = `DISASSOCIATED`;

Figure 7: Node operations in FIMF.

H: the set of requesting nodes;
P: the ferry route in the `WORKING` mode;
mode: which mode the node is in;
1. `IDLE` mode
    Set *H* to be an empty set;
    Move on a default route;
    Broadcast location messages periodically;
    On reception of a `Service_Request` message from node *i*:
        Add *i* to *H*;
        Compute new ferry route *P*;
        mode = `WORKING`;
    On reception of messages from a node:
        Exchange messages with the node;
2. `WORKING` mode
    Move along route *P*;
    Broadcast location messages periodically;
    On reception of a Request/Update message from node *i*:
        Add *i* to *H*;
        Compute new ferry route *P*;
    On finish of transmission or meeting with node *j*:
        Remove *j* from *H*;
        IF *H* is not empty,
            Compute new ferry route *P*;
        ELSE
            mode = `IDLE`;
    On reception of messages from a node:
        Exchange messages with the node;

Figure 8: Ferry operations in FIMF.

ferry maintains a set of nodes H that have requested service and tries to meet these nodes to relay messages. The ferry trajectory control mechanism will be discussed in Section 4.3. When a request is received, the ferry updates H, computes a new ferry route and adjusts its movement to follow the new route. The ferry also recomputes its route when a `Location_Update` message is received. When the ferry arrives at the location of a node reported in its request or update messages, the ferry assumes that it has finished the visit with the node and removes it from H. The ferry may also meet a node in H, say node i, when the ferry is on its way to meet another node, in which case the ferry assumes it has visited node i and removes node i from H. After updating H, the ferry recomputes its route and moves on the new route. When H becomes empty which means the ferry has visited all requesting nodes, it returns to the default route and enters the `IDLE` mode. In both modes, when the ferry comes close to a node, the ferry may exchange messages with the node.

When in the `IDLE` mode, the ferry moves on a default route and waits for requests from nodes. Since the transmission range of nodes' long range radios may be limited due to energy constraints, a node must be close enough to the ferry in order to send a request to the ferry. So the default ferry route should be designed to maximize the chance that the ferry is close to nodes. Given nodes are mobile and their movement is unknown in advance, it is difficult, if not impossible, to design an optimal ferry route. In this work, we adopt a simple approach in which the area is divided into a grid of square cells and the ferry route is designed to "scan" through the cells in a row-by-row order. Specifically, the ferry route moves forward through the first row of cells, then moves backward on the second row and repeats this back and forth pattern until all cells have been visited, whereupon, the route returns to the originating cell, forming a closed route. The cell size is chosen to be no larger than $\sqrt{2}R_l$ where R_l is the transmission range of nodes' long range radios. So when the ferry moves through the route, nodes at every location in the area have a chance to send messages to the ferry.

The operations of FIMF differ from NIMF in the proactive movement of the ferry, instead of nodes, and the use of long range radios in nodes to transmit control messages. However, message forwarding, device discovery and message drop computation are the same in both schemes.

4.2 Node Notification Control

In FIMF, nodes send notification messages to request communication service from the ferry using long range radios. Notification messages can be either a `Service_Request` message or a `Location_Update` message. A `Service_Request` message indicates the node's intent to communicate with the ferry while a `Location_Update` message informs the ferry about the node's new location. Both messages include the node's current location. Because transmission over long distance is expensive in energy consumption, the goal of *notification control* is to minimize message drops while considering energy constraints.

To control the transmission of notification messages, we consider the following factors: message drops, ferry location and energy consumption. As for message drops, we adopt a similar policy as in NIMF. Specifically, a node sends a request message to the ferry only when Eq. (1) is true. The only difference is in the computation of t_d, the time slot in which the node is expected to meet the ferry.

In FIMF, t_d is determined by the ferry movement which is unknown to nodes. In this paper, we estimate t_d as $t_0 + ct_f$ where t_0 is the current time slot, t_f is the latency for the ferry to move directly to meet the node, and c is a constant.

We also consider the ferry's location in notification control. Let d_f be the distance from the node to the ferry. Let R_l be the transmission range of nodes' long range radios. In FIMF, a node sends a request to the ferry only when $d_f < \gamma$ where γ is a system parameter and $\gamma < R_l$. We use γ to reduce the chance that the ferry moves out of the node's transmission range after the node has sent a `Service_Request` message.

We now turn to the energy consumption issue. To achieve certain node or network life time, nodes may have some energy usage constraints, which are generally application specific. In this paper, we use a simplified model which limits the transmission of notification messages. We define *notification message rate* (NMR) as the average number of notification messages sent per second. To enforce energy constraints, a node i is allowed to send a notification message only when its NMR v_i is below a predefined threshold λ.

By combining the above factors, we determine a policy that states that a node transmits a `Service_Request` message only when Eq. (1) is true, $d_f < \gamma$ and $v_i < \lambda$.

Notification control also determines when `Location_Update` messages are sent to inform the ferry about the node's new location. Suppose that the short range radios have transmission range of R_s. If a node has moved but is still within R_s distance from the location it reported to the ferry, the node does not need to send an update message to the ferry. This is because when the ferry moves to the reported location, the ferry can still communicate with the node. In addition, for the ferry to successfully receive the notification messages, the distance between the node and the ferry must be small enough, i.e., $d_f < R_l$. So a node sends a `Location_Update` message only when $d_n > R_s$, $d_f < R_l$ and $v_i < \lambda$ where d_n is the node's distance to the location it reported to the ferry.

4.3 Ferry Trajectory Control

In this section, we discuss how the ferry controls its trajectory to meet nodes with the goal of minimizing message drops. Before describing the *trajectory control* mechanism, we first define the ferry route problem. Suppose P is a route that starts from the ferry's location and visits all nodes that have sent requests to the ferry. Assume that nodes remain in their locations, the ferry can compute the latency before it visits each node in route P given the location of these nodes and the ferry speed. Let s_i be the latency for node i. Let $D_i^n(t)$ be the message drop rate in node i during time slot t, and $D_i^f(t)$ be the drop rate in the ferry for destination i during slot t.[4] We define the expected message drops for route P as

$$D^P = \sum_{i=1}^{k} \sum_{l=0}^{s_i} (D_i^n(t_0 + l) + D_i^f(t_0 + l)) \quad (3)$$

where k is the number of requesting nodes and t_0 is the current time slot. So the ferry route problem can be stated as finding the route that minimizes D^P. When $D_i^n(t) + D_i^f(t) = 1$ for all i and t, this problem becomes a Minimum Latency Problem (MLP) [6] where D^P can be interpreted as the sum of latency for the ferry to visit each node. MLP is known to be NP-hard, so this problem is as well. In addition, since nodes are mobile, D^P is only a rough estimation of message drops. Thus we seek to develop heuristics for ferry route design.

[4] $D_i^n(t)$ can be measured in node i and transmitted to the ferry in `Service_Request` or `Location_Update` messages.

We study the following two heuristics. The first one is the nearest neighbor (NN) heuristic in which the ferry always visits the closest node after it finishes meeting with a node. The second heuristic is a traffic-aware (TA) heuristic which considers both location and message drop information. The TA heuristic is based on local optimization technique 2H-opt used in the traveling salesman problem (TSP) [5]. However, instead of optimizing the length of the route as in TSP, the TA heuristic tries to optimize the expected message drops D^P.

After computing a route, the ferry adjusts its movement to follow the new route. Since nodes are mobile, there might be chances that the ferry might miss the node it is trying to visit. Under these situations, when the ferry moves to the location reported by the node, the ferry assumes that it has finished the visit with the node and recomputes its route to meet with remaining nodes. So the operations of the ferry are not affected. Because we assume that the ferry moves faster than nodes, the probability of these misses would be small.

5. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the Message Ferrying schemes through *ns* simulations. We first describe our simulation implementation, performance metrics and methodology in Section 5.1. Then we evaluate both the node-initiated and ferry-initiated MF schemes and compare their performance with that of Epidemic routing [33]. The results confirm that the MF schemes are efficient in both data delivery performance and energy consumption in sparse mobile ad hoc networks.

5.1 Metrics and Methodology

We implement both the NIMF and FIMF schemes in *ns* simulator. For short range communication, we use 802.11 DCF as MAC layer and the default energy model provided in *ns*, i.e., 250m transmission range and 0.282W transmit power. For long range communication, we choose not to simulate a specific MAC protocol in detail, but instead use a simplified model in which data transmission has no loss or delay. This simplification will not affect our results because long range radios are only used for low rate control messages, thus MAC contention is not a major concern. In addition, since normal transfer delay is on the order of minutes or even hours, we can ignore transmission delay when using long range radio. We only use transmission power in regular nodes for energy consumption computation. Energy consumption at the ferry is not counted because the ferry is not limitated in power supply. In general transmit power for distance d is proportional to d^k where k is the path loss exponent [28]. In our simulations, k is set to 4 and the computation of transmit power is based on distance from the transmitting node to the ferry. We compare the MF schemes with Epidemic routing whose performance acts as a baseline.

We use both data delivery and energy metrics to evaluate the performance of the MF schemes. The *message delivery rate* is defined as the ratio of the number of successfully delivered messages to the total number of messages generated. The *message delay* is defined as the average delay between the time a message is generated and the time the message is received at the destination. In Epidemic routing, because multiple copies of a message may be received at the destination, the message delay is computed based on the earliest time a message is received. The message delivery rate and message delay reflect how efficient data delivery is. To measure energy efficiency, we use the *delivered messages per unit energy* which is defined as the average amount of data delivered per unit energy consumption. This measures the efficiency of energy usage.

We use the following default settings in our simulations, un-

Scheme	Parameter	Value
NIMF	WTP threshold (ω)	0.9
FIMF	Transmission range of nodes' long range radio (R_l)	2000
	Distance threshold for request transmission (γ)	1800
	NMR threshold (λ)	0.001
Both	Avg time between a node's visits with the ferry (α)	1000
	Message drop threshold (β)	0.3

Table 1: Default parameter settings

less specified otherwise. Each simulation run has 40 nodes on a $5000m \times 5000m$ area. 25 nodes are randomly chosen as sources which send messages to randomly chosen destinations every 20 seconds. Messages are of size 500 bytes and the timeout value is 8000sec. Nodes move in the area according to the random-waypoint model [17] with a maximum speed 5m/s and pause time 50sec. The node buffer size is 400 messages. In the NIMF and FIMF simulations, a single ferry is used and the ferry speed is 15m/s. The default ferry route follows a rectangle with (1250,1250) and (3750,3750) as diagonal points (we use a 2×2 grid in computing the default route for FIMF). In FIMF, nodes' long range radios transmit at 500kbps. The default settings for other parameters are listed in Table 1.

5.2 Impact of node buffer size

We first evaluate the impact of node buffer size on data delivery performance. We simulate the following schemes, NIMF, FIMF with the NN heuristic (FIMF-NN), FIMF with the TA heuristic (FIMF-TA) and Epidemic routing (ER). Fig. 9(a) shows the message delivery rate under different node buffer sizes. NIMF and both FIMF schemes significantly outperform Epidemic routing for all buffer sizes. For example, when buffer size is 200 messages, all MF schemes achieve more than 81% delivery rate while Epidemic routing only delivers 20% of messages. This is because in the MF schemes the proactive movement of nodes or the ferry increases connectivity among nodes, leading to higher message delivery rate. In addition, the MF schemes avoid the buffer contention problem caused by flooding in Epidemic routing. As the node buffer size increases, the message delivery rate for Epidemic routing also increases but is still lower than the MF schemes.

Fig. 9(b) shows the message delay which tends to increase as the buffer size. For Epidemic routing, the increase of delay is because messages can stay longer in buffer before being purged out by new messages. For MF, as the buffer size increases, a node can buffer more messages before transmitting to the ferry, thus leading to increased delay. Epidemic routing achieves much lower delay as compared to the MF schemes. This is because MF *explicitly* delays message delivery by batching messages in nodes or the ferry, either to increase a node's work time percentage (in NIMF) or to reduce energy consumption of long range communication (in FIMF).

Figure 9(c) presents the energy efficiency metric. The MF schemes achieve better energy efficiency, by 8 to 30 times, than Epidemic routing. So using the same amount of energy, the MF schemes can deliver more than 8 times as many messages. Such great reduction in energy consumption comes from the fact that messages are forwarded in 2 hops to reach their destinations in MF. In contrast, Epidemic routing floods messages throughout the network, resulting in a large number of redundant message transmissions. As the buffer size increases, the FIMF schemes improve their energy efficiency. This is because with a larger buffer, a node can reduce the transmission of request messages, thus reducing the overhead of long range communication. For NIMF and Epidemic routing, the increase of buffer size does not improve energy efficiency. In all cases NIMF achieves better energy efficiency than the FIMF schemes because nodes in NIMF do not need to send control messages to the ferry.

The above simulations use constant bit rate (CBR) traffic. We also conduct experiments with bursty traffic. The MF schemes achieve very similar performance under bursty traffic as compared to CBR traffic. The only difference is a slight decrease in message delivery rates. Due to space limitation, we do not present these results in this paper. Both FIMF schemes achieve similar performance in message delivery rate, delay and energy efficiency in all these simulations. So in the rest of this section, we will only present the results for FIMF-TA.

5.3 Impact of node mobility

In this section we study how node mobility affects each scheme. We simulate three mobility models. In the random-waypoint (RW) model, a node randomly picks a destination within the area and moves toward the destination with a speed uniformly distributed between 0 and a maximum speed s_{max}. The second mobility model, limited random-waypoint (LRW) model, is a variation of the RW model in which when a node chooses a destination, it picks a location within the $400m \times 400m$ area centered at its location. So this model tries to limit the distance of each move. The third mobility model is the area-based (AB) model. In this model, 10 nodes are moving according to the RW model while the other nodes move within a randomly chosen $400m \times 400m$ area. This model reflects situations where most nodes tend to move within a small area. We simulate different s_{max} at 5m/s and 10m/s.

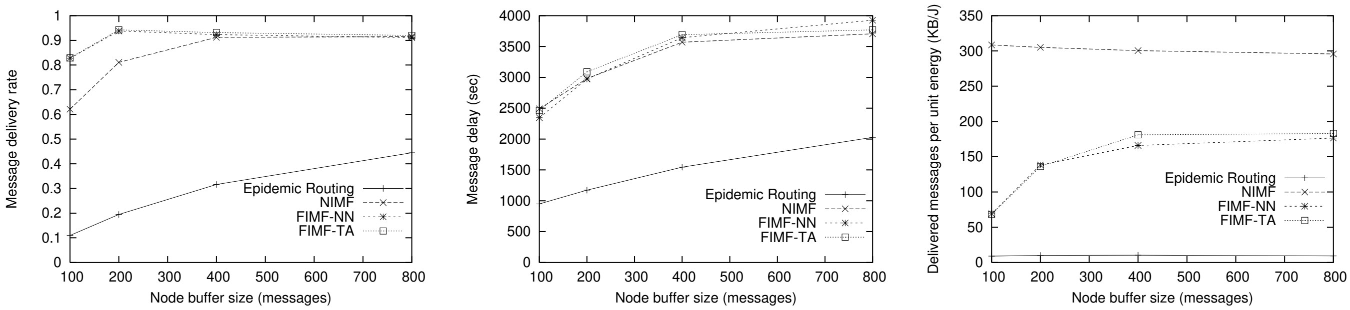
Table 2 summarizes the performance results. The MF schemes outperform Epidemic routing significantly in all scenarios we study. For example, when s_{max} is 10m/s, both NIMF and FIMF achieve message delivery rate of more than 84% while Epidemic routing achieves less than 44%. The MF schemes also obtain much better energy efficiency.

We make two observations. First, while node mobility has significant impact on the performance for Epidemic routing, both NIMF and FIMF are much less affected by node mobility. This is because in Epidemic routing, nodes rely on their movement to meet others. So mobility characteristics greatly affect data delivery performance. For example, Epidemic routing performs better when nodes move globally or at a higher speed. In NIMF, nodes take proactive movement to meet with the ferry so data delivery performance is less affected by regular node mobility. A similar argument also applies to FIMF. Second, the performance of NIMF is affected by node speed. This is because nodes need to proactively move to meet the ferry. Due to the WTP constraint, nodes with a higher speed are able to visit the ferry more frequently because they spend less time on each detour to meet the ferry. In contrast, FIMF is insensitive to node speed because it is the ferry who takes proactive movement.

5.4 Impact of WTP threshold on NIMF performance

In this section, we evaluate how the WTP threshold setting affects the NIMF scheme. The WTP threshold ω controls how much time a node is allowed for proactive movement. Fig. 10(a) shows the message delivery rate when the message timeout are 3000 and 8000 seconds. As the WTP threshold ω increases, the message delivery rate decreases since nodes visit the ferry less frequently. The delivery rate also decreases for smaller message timeout values because more messages would be dropped due to timeout.

Fig. 10(b) presents the message delay which tends to be unaffected by the setting of ω. The reason is that in NIMF, a node explicitly batches outgoing messages and tries to delay its visit with



(a) Message delivery rate

(b) Message delay

(c) Delivered messages per unit energy

Figure 9: Performance under different node buffer sizes.

Mobility Model (speed)	Scheme	Delivery Rate	Delay (sec)	Energy efficiency (KB/J)
Random Waypoint (0 - 10m/s)	NIMF	0.953	3490	306
	FIMF	0.938	3530	172
	ER	0.437	1039	10
Limited Random Waypoint (0 - 10m/s)	NIMF	0.893	3918	289
	FIMF	0.842	4145	129
	ER	0.127	2076	9
Area Based (0 - 10m/s)	NIMF	0.842	3739	288
	FIMF	0.883	3838	144
	ER	0.155	1476	9
Random Waypoint (0 - 5m/s)	NIMF	0.912	3569	300
	FIMF	0.931	3691	181
	ER	0.316	1546	10
Limited Random Waypoint (0 - 5m/s)	NIMF	0.699	3896	267
	FIMF	0.850	4091	137
	ER	0.061	2221	6
Area Based (0 - 5m/s)	NIMF	0.731	3862	271
	FIMF	0.841	4036	137
	ER	0.112	2239	8

Table 2: Performance under different node mobility

the ferry as much as possible. So the message delay remains the same under different WTP thresholds.

Fig. 10(c) shows the energy efficiency metric. For both message timeout values, energy efficiency decreases as ω increases. This is because of message drops in the ferry. As ω increases, nodes are more restricted in their movement to meet the ferry and more messages will be dropped in the ferry because of timeout. So energy used to transmit these dropped messages from nodes to the ferry is wasted, reducing energy efficiency. NIMF achieves better energy efficiency for larger message timeout values because of fewer message drops.

5.5 Impact of NMR threshold on FIMF performance

In this section we study how FIMF performs under different NMR thresholds λ. The use of λ is to limit energy consumption of nodes' long range communication. Fig. 11(a) shows the message delivery rate when the message timeout are 3000 and 8000 seconds. Note that the x axis in Fig. 11 is the average node notification interval which equals $1/\lambda$. For 8000 second timeout, the delivery rate remains the same under different λ because the message timeout is relatively large, so nodes can delay the transmission of request messages without resulting in message timeout. For 3000 second timeout, the delivery rate decreases as the notification interval increases because nodes are more restricted in sending out notification messages, resulting in more message drops.

Fig. 11(b) presents the message delay which tends to be independent of λ. This is because FIMF explicitly delays message transmission for energy saving in long range communication. Fig. 11(c) shows the energy efficiency metric. For both message timeout values, energy efficiency increases as the average node notification interval. This is because with a larger notification interval, nodes wait for longer time before requesting service from the ferry and transmit more messages on each visit with the ferry. So the transmission overhead for notification messages is reduced, resulting in higher energy efficiency. For the same reason, FIMF achieves better energy efficiency in simulations with larger timeout values.

5.6 Impact of transmission range on FIMF performance

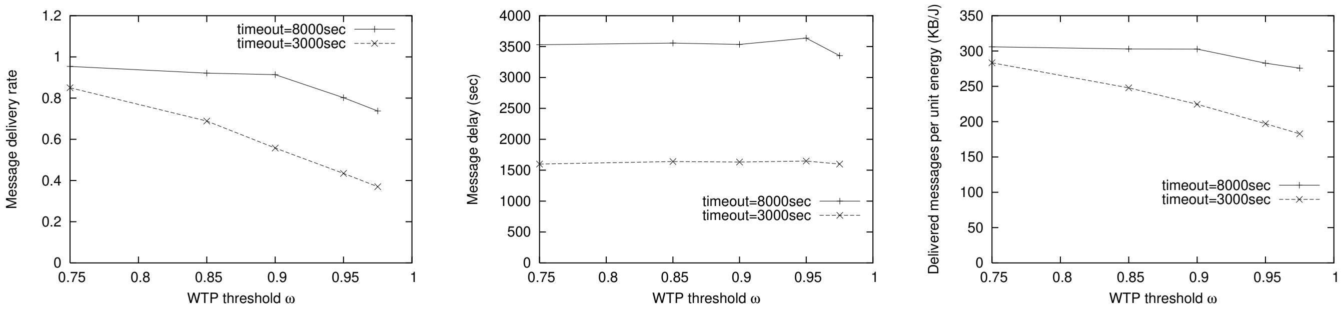
We now study how the transmission range of nodes' long range radios affects the performance of FIMF. We simulate with message timeout of 3000 and 8000 seconds. Fig. 12(a) shows the message delivery rate which increases as the transmission range R_l, especially when R_l is small. This is because with a larger R_l, the default route will be shorter and the ferry takes less time to finish one round of the route. So the ferry will be in a node's range more frequently, resulting in better message delivery rate.

Fig. 12(b) shows the message delay which is unaffected by R_l. Fig. 12(c) plots the energy efficiency metric. Energy efficiency first increases when the transmission range R_l is small and then drops when R_l becomes large. The reason is for small R_l, the improvement of energy efficiency stems from the increase in the message delivery rate and the reduction in message drops, especially for the case with 3000 second timeout. When fewer messages are dropped, less energy is wasted in transmitting messages to the ferry which will be dropped later, improving energy efficiency. For large R_l, the improvement in message drops is small as R_l increases. Energy used to transmit notification messages to the ferry becomes the dominant factor. As R_l increases, more energy will be used in the transmission of notification messages, resulting in lower efficiency.

6. DISCUSSION

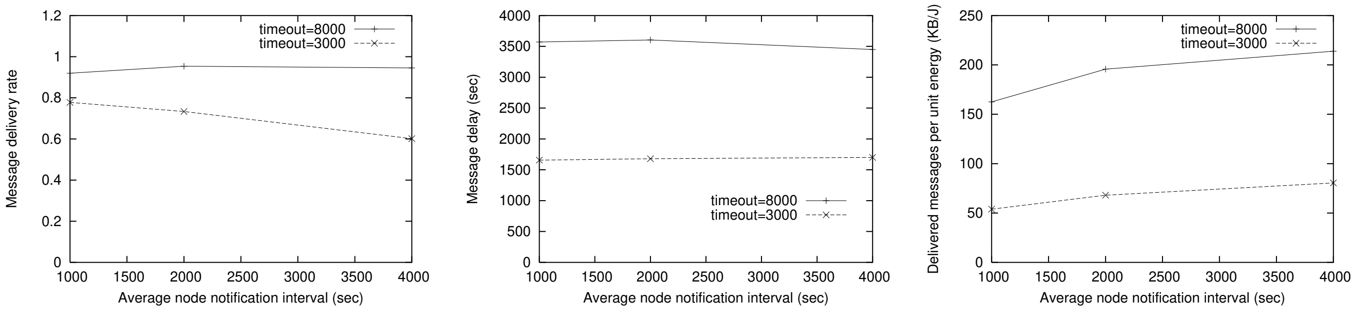
In this section, we will discuss some related design issues in the Message Ferrying schemes.

Multiple Ferries. In this paper, we have focused on the use of a single message ferry to provide communication capability. We expect the MF schemes can be easily extended to the case with multiple ferries. Multiple ferries can potentially improve message transport capacity and robustness against ferry failures. With multiple ferries in deployment, there is also flexibility in relaying messages



(a) Message delivery rate (b) Message delay (c) Delivered messages per unit energy

Figure 10: Performance of the NIMF scheme under different WTP thresholds.



(a) Message delivery rate (b) Message delay (c) Delivered messages per unit energy

Figure 11: Performance of the FIMF scheme under different NMR thresholds.

and balancing load among ferries. We are currently studying the issues of cooperative routing with multiple ferries. Specifically, how is the movement of ferries and nodes organized to improve physical connectivity among nodes? And given the ferry movement, how are messages forwarded to improve data delivery performance and energy efficiency?

Contention. In the MF schemes, messages are relayed via the ferry. This may create *transmission contention* among nodes when multiple nodes try to communicate with the ferry simultaneously. When the ferry has only limited buffers, there is also *buffering contention* among messages from different nodes. In this paper, we consider situations where all messages are of the same importance. Transmission contention is mainly resolved by the MAC protocol used and there is no buffer contention. In more realistic situations especially in crisis scenarios, given the limited connectivity between nodes and the ferry, it would be important to schedule transmission such that application performance can be maximized. This requires some information about the content of messages such as message priority. A simple priority-based approach is as follows. Each message is tagged with a priority number when generated. When a node detects the ferry is in range, the node informs the ferry about the number of buffered messages and their priorities. The ferry then calculates a transmission schedule based on message priority. Following the transmission schedule, the ferry will either transmit messages to a node or polling a node which is allowed to send its messages to the ferry after the poll message. In this way, the ferry has complete control about how message exchange occurs and reduces contention between nodes. In the FIMF scheme, the ferry has extra flexibility of controlling the contact period with regular nodes. We defer the study of these issues to future work.

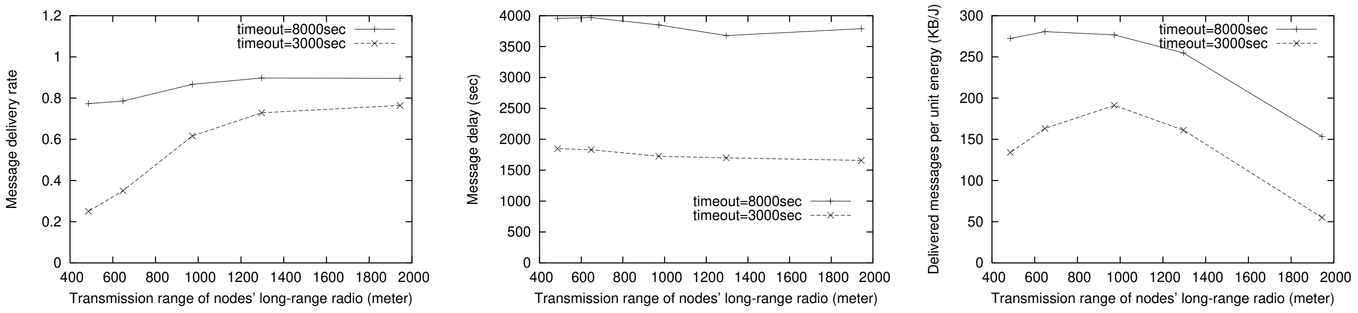
Coordination among Regular Nodes. While this paper considers the case where nodes operate independently, regular nodes can coordinate with each other for data delivery. For example, regular nodes can form connected *clusters*. Within a cluster, one or more *gateway nodes* are in charge of communicating with the ferries. Other nodes communicate with these gateway nodes using traditional ad-hoc network routing protocols.

Long Range Communication. In the MF schemes, the ferry broadcasts control messages across the deployment area using long range communication. This does not pose much limitation because the overhead is not significant due to the small size of control messages and the low broadcasting frequency. In addition, the MF schemes apply to situations where the ferry has a shorter radio range.

In this paper, we focus on data delivery using the ferries. In networks where direct communication among nodes is feasible, it may be desirable to transmit messages using long range communication, even with higher energy cost. For example, direct communication can be used when messages have stringent delay requirements. It would be interesting to study approaches that use both direct communication and the MF approach to balance energy consumption and message delay. We leave this as a topic of future work.

7. RELATED WORK

In this section, we review related work on sparse networks and mobility-assisted schemes. Ad hoc routing has been an active research field in recent years and many routing algorithms have been developed, such as DSR [17], DSDV [25], AODV [26], GPRS [19], Zone routing [16], LAR [20] and CEDAR [30]. All these routing algorithms consider connected networks where an end-to-end path exists between two nodes in the network. Our work in this paper differs significantly in its focus on sparse networks.



(a) Message delivery rate (b) Message delay (c) Delivered messages per unit energy

Figure 12: Performance of the FIMF scheme under different transmission ranges.

Sparse Network Architectures. Goodman *et al.* [12] propose the *Infostation* architecture in which wireless ports called Infostations are geographically distributed to provide high bit-rate connections in the their vicinities. Infostations could be placed at accessible locations such as airport and building entrances, and do not offer continuous coverage. So the Infostation architecture supports "many-time many-where" communication.

Shah *et al.* [29] propose DataMULE, an architecture for collecting data in sparse sensor networks. DataMULE uses mobile entities in the environment to transport data from sensor nodes to access points. DataMULE aims to achieve energy saving in sensor nodes by using short range radios. DataMULE is similar to the MF schemes in its use of mobile entities to deliver data. However, its focus on static sensors and data collection applications differs from the MF schemes. In addition, the MF schemes explicitly utilize non-random or controlled movement to deliver data.

Fall [11] proposes a *Delay-Tolerant Network* (DTN) architecture to interconnect challenged networks such as sensor networks and interplanetary networks. In many challenged networks, end-to-end paths might not exist between nodes. DTN is based on asynchronous message forwarding paradigm and focuses on how to achieve interoperability among different challenged networks. Our work complements [11] in that MF addresses how data is delivered in one specific kind of challenged networks, sparse mobile ad hoc networks.

Ahmed *et al.* [2] introduce the use of a *range extension* infrastructure to overcome partitions in sparse MANETs. A range extension infrastructure consists of airplanes or satellites which communicate with gateway nodes in each cluster of the ground ad hoc networks. While this scheme employs proactive gateway motion for communication purposes, it differs from our MF schemes in its reliance on the range extension infrastructure and its maintenance of persistent connectivity between nodes.

Small and Haas [31] propose SWIM, a network architecture that combines the Infostation architecture and the ad hoc networking model. Specifically, by replicating and spreading data throughout the mobile nodes in the network, this scheme can significantly reduce the delay until one of th replicas reaches an Infostation.

Our previous work [35] first introduces the Message Ferrying concept and studies its use in networks with stationary nodes. We develop efficient algorithms to control ferry trajectory to minimize delay while meeting nodes' data rate requirements.

Mobility-Assisted Schemes. The use of mobility to overcome network partitions has been considered for data delivery in sparse ad hoc networks. In general, two kinds of schemes have been proposed, namely reactive schemes and proactive schemes. In *reactive* schemes, applications rely on movement that is inherent in the devices themselves to help deliver messages, while in *proactive* schemes, devices move proactively and specifically in order to communicate with others.

Epidemic routing, proposed by Vahdat and Becker [33], is a flooding-style scheme in which nodes propagate messages to all nodes they meet. This scheme is simple and very robust to network partitions and node failure. The disadvantage is that it generates a large number of redundant messages, leading to poor scalability and high energy cost when nodes have limited buffers and energy supplies. Davis *et al.* [9] proposes an improved scheme over Epidemic routing by exploiting node mobility statistics. Nodes estimate their probabilities of meeting other nodes in the future and drop messages selectively when buffers overflow. In the work of [23], Nain *et al.* propose a Mobile Relay Protocol to take advantage of node mobility to overcome partitions for message delivery.

The above schemes are *reactive* schemes in that when disconnected, nodes passively wait for their own mobility to allow them to re-connect. Because encounters between nodes can be unpredictable and rare, these schemes suffer potentially low data delivery rates and large delays. To address these problems, Li and Rus [21] propose a proactive scheme in which mobile nodes actively modify their trajectories in order to transmit messages as soon as possible. The authors propose an algorithm to compute an optimal trajectory for relaying a message among nodes. However, it is difficult to extend this algorithm to support multiple message transmissions simultaneously.

Mobility-assisted schemes have been proposed to use in a variety of applications. In the ZebraNet project [18], sensors are attached to zebras and used to study the behavior of wildlife. Due to device form factor and energy constraints, only short range radios can be used. As zebras move, these sensors form a sparse network and are not able to transfer data to users. Users need to move close to the zebras, by driving a car or a plane, to collect data from these sensors. In the DakNet project [1], vehicles are used to transport data between remote areas such as villages and cities to provide store-and-forward Internet access. In the work of [4], Beaufour *et al.* investigate sparse sensor networks and propose to leverage the movement of mobile individuals to disseminate data between disconnected sensors. Zhen *et al.* [8] study the problem of forming ad hoc relaying networks over moving vehicles on highways and show via simulations that vehicle mobility can contribute to successful message delivery and reduced delay. Morris *et al.* [22] describe CarNet, a scalable ad hoc network system on cars. It is based on geographic forwarding and Grid location service to achieve scalability. CarNet can be used for applications like traffic congestion monitoring and fleet tracking. There is also other work on exploiting mobility for security [7], location [14] and routing [10, 32].

Capacity of Wireless Networks. There is some theoretical work on the capacity of mobile ad hoc networks. In a seminal paper, Gupta and Kumar [15] study a model of ad hoc networks with fixed nodes and show that when the number of nodes per unit area n increases, the per node throughput decreases as $O(\frac{1}{\sqrt{n}})$. Grossglauser and Tse [13] show that with loose delay constraints, node mobility can dramatically improve network capacity. They prove that the per node throughput can be kept constant as the number of nodes per unit area increases. The improvement of throughput comes at the price of increased delay. In the work of [3, 24], the authors study the issue of delay and capacity in mobile networks.

Topology Control. There is also some research on *topology control* in wireless multi-hop networks [27, 34]. This body of work focuses on adjusting the transmit powers of nodes in a multihop wireless network in order to create a topology with desired properties, e.g., maintaining network connectivity. The topology control approach assumes that the transmit power can be arbitrarily adjusted and normally tries to minimize the power used. In addition, the topology control approach maintains constant network connectivity. In contrast, the MF scheme relies on node mobility to provide interim, yet regular, connectivity.

8. CONCLUSION

In this paper we have studied the problem of efficient data delivery in sparse mobile ad hoc networks and presented a Message Ferrying approach to address this problem. MF is a mobility-assisted approach which utilizes a set of special mobile nodes called message ferries to provide communication service for nodes in the area. The main idea behind the MF approach is to introduce non-randomness in the movement of nodes and exploit such non-randomness to help deliver data. We develop two variations of the MF scheme, depending on whether ferries or nodes initiate proactive movement. We have evaluated the performance of MF on a variety of network conditions. Our simulation results show that the MF approach is very efficient in both data delivery and energy consumption. For example, the MF schemes deliver more messages (by more than 45% of all messages) and achieve higher delivered messages per unit energy (by more than 7.5 times) than Epidemic routing.

Our future work will include addressing the issues discussed in Section 6 as well as completion of a prototype system using PDAs as nodes and implementing a ferry in a campus shuttle bus.

9. REFERENCES

[1] DakNet project. http://web.media.mit.edu/˜amir/daknet/.

[2] M. Ahmed, S. Krishnamurthy, R. Katz, and S. Dao. Trajectory control of mobile gateways for range extension in ad hoc networks. *Computer Networks: The International Journal of Computer and Telecommunication Networking (COMNET)*, to appear.

[3] N. Bansal and Z. Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *IEEE INFOCOM'2003*, April 2003.

[4] A. Beaufour, M. Leopold, and P. Bonnet. Smart-tag based data dissemination. In *First ACM WSNA Workshop*, September 2002.

[5] J. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing*, 4:387–411, 1992.

[6] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. In *ACM Symposium on Theory of Computing*, pages 163–171, 1994.

[7] S. Capkun, J. Hubaux, and L. Buttyan. Mobility helps security in ad hoc networks. In *ACM MobiHoc*, June 2003.

[8] Z. Chen, H. Kung, and D. Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *MobiHoc 2001*, October 2001.

[9] J. Davis, A. Fagg, and B. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In *International Symposium on Wearable Computing*, October 2001.

[10] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In *ACM MobiHoc*, June 2003.

[11] K. Fall. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM'03*, 2003.

[12] D. Goodman, J. Borras, N. Mandayam, and R. Yates. INFOSTATIONS: A new system model for data and messaging services. In *IEEE VTC'97*, volume 2, pages 969–973, May 1997.

[13] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *IEEE INFOCOM 2001*, April 2001.

[14] M. Grossglauser and M. Vetterli. Locating nodes with EASE: Mobility diffusion of last encounters in ad hoc networks. In *IEEE INFOCOM'2003*, April 2003.

[15] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46:388–404, March 2000.

[16] Z. Haas and M. Pearlman. The performance of query control schemes for the zone routing protocol. In *ACM SIGCOMM*, 1998.

[17] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks. In *ACM SIGCOMM*, August 1996.

[18] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *ASPLOS'02*, October 2002.

[19] B. Karp and H. Kung. Greedy perimeter stateless routing for wireless networks. In *ACM MobiCom 2000*, August 2000.

[20] Y. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *ACM MOBICOM*, November 1998.

[21] Q. Li and D. Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *ACM MOBICOM*, August 2000.

[22] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto. CarNet: A scalable ad hoc wireless network system. In *9th ACM SIGOPS European Workshop*, Sept. 2000.

[23] D. Nain, N. Petigara, and H. Balakrishnan. Integrated routing and storage for messaging applications in mobile ad hoc networks. In *WiOpt 2003*, March 2003.

[24] E. Perevalov and R. Blum. Delay limited capacity of ad hoc networks: Asymptotically optimal transmission and relaying strategy. In *IEEE INFOCOM'2003*, April 2003.

[25] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *Computer Communication Review*, 24, October 1994.

[26] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

[27] R. Ramanathan and R. Hain. Topology control of multihop wireless networks using transmit power adjustment. In *IEEE INFOCOM*, 2000.

[28] T. S. Rappaport and T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2001.

[29] R. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, 2003.

[30] R. Sivakumar, P. Sinha, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17:1454–1465, August 1999.

[31] T. Small and Z. Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way). In *ACM MobiHoc*, June 2003.

[32] W. Su, S. Lee, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11:3–30, 2001.

[33] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. *Technical report, Duke University*, 2000.

[34] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *IEEE INFOCOM 2001*, April 2001.

[35] W. Zhao and M. Ammar. Proactive routing in highly-partitioned wireless ad hoc networks. In *the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems*, May, 2003.