Logarithmic Store-Carry-Forward Routing in Mobile Ad Hoc Networks

Jie Wu, *Senior Member*, *IEEE*, Shuhui Yang, *Student Member*, *IEEE*, and Fei Dai

Abstract—Two schools of thought exist in terms of handling mobility in mobile ad hoc networks (MANETs). One is the traditional connection-based model, which views node mobility as undesirable and tries to either remove (through recovery schemes) or mask (through tolerant schemes) the effect of mobility. The other is the mobility-assisted model, which considers mobility as a desirable feature, where routing is based on the store-carry-forward paradigm with random or controlled movement of mobile nodes (called ferries). It is well known that mobility increases the capacity of MANETs by reducing the number of relays in routing. Surprisingly, only two models, diameter hop count in the connection-based model and constant hop count in the mobility-assisted model, which correspond to two extremes of the spectrum, have been systematically studied. In this paper, we propose a new routing model that deals with message routing, as well as trajectory planning, of the ferries that carry the message. A logarithmic number of relays are enforced to achieve a good balance among several contradictory goals, including increasing network capacity, increasing ferry sharing, and reducing moving distance. The model considers the dynamic control of ferries in terms of the number of ferries, trajectory planning of ferries, and node communication and synchronization. The effectiveness of the proposed model is evaluated analytically, as well as through simulation.

Index Terms—MANETs, mobile nodes, network capacity, store-carry-forward, trajectory planning.

1 Introduction

Mobile ad hoc networks (MANETs) are characterized by their node mobility and lack of infrastructure. The routing process in MANETs, one of the most important functions at the network layer, has been extensively studied in recent years. However, mobility management in MANETs is still relatively understudied. The main issue centers around *whether mobility should be treated as a foe (undesirable) or a friend (desirable)*.

The traditional *connection-based model* used in MANETs, including the existing protocols (DSR, AODV, and ZRP), is built on the premise that the underlying network is connected and views node mobility as undesirable. In this model, the source and destination are connected through a path in a connected graph representing the network. However, mobility is treated as a side issue through a simple *recovery scheme*. For example, a route disruption caused by node movement is dealt with by either route rediscovery or a local fix in a typical reactive approach. More recently, mobility has been identified as a serious threat to the traditional model [1]. The threat is mainly caused by asynchronous sampling of Hello messages and various protocol delays that result in an inconsistent global state. Several *tolerant schemes* have been proposed [1], [2] as the first attempt to mask the effect of node movement and to construct a consistent global state for various applications.

Grossglauser and Tse [3] showed that mobility increases the capacity of MANETs by restricting the number of relays to a constant as opposed to $O(\frac{1}{\log n})$ [4] capacity in the connection-based model, where n is the number of nodes. Inspired by this result, *movement-assisted routing* tries to exploit node movement for the routing process. In the store-carry-forward paradigm, a mobile node (called a *ferry* [5]) first stores the routing message from the source, carries it while in motion, and then forwards it to an intermediate node or the destination. This model supports routing in an unconnected graph by virtual connectivity through node movement. Movement-assisted models can be classified based on random (uncontrolled) movement, such as epidemic routing (ER) [6], and controlled movement, such as message ferrying (MF) [5], [7]. In controlled movement, various design issues exist, including the number of ferries, trajectory planning of ferries, and node communication and synchronization.

Although extensive work has been conducted on both models, relatively little work has been done on controlling the number of relays or the average hop count in a routing process. In the traditional connection-based model, the average hop count grows with the spatial diameter of the network (simply called diameter hop count), that is, $O(\sqrt{n})$. At the other extreme of the spectrum, most controlled-movement-assisted schemes use a constant number of relays (simply called constant hop count); however, these schemes incur a long moving distance (and delay) in message forwarding. Most random movement-assisted schemes do not control the hop count to ensure delivery. Although hop count can be shortened through replication (since the message is likely to reach the destination quicker),

- J. Wu and S. Yang are with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431. E-mail: jie@cse.fau.edu, syang1@fau.edu.
- F. Dai is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58105. E-mail: fei.dai@ndsu.edu.

Manuscript received 19 Apr. 2006; revised 18 Aug. 2006; accepted 20 Sept. 2006; published online 9 Jan. 2007.
Recommended for acceptance by S. Olariu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0098-0406.
Digital Object Identifier no. 10.1109/TPDS.2007.1061.

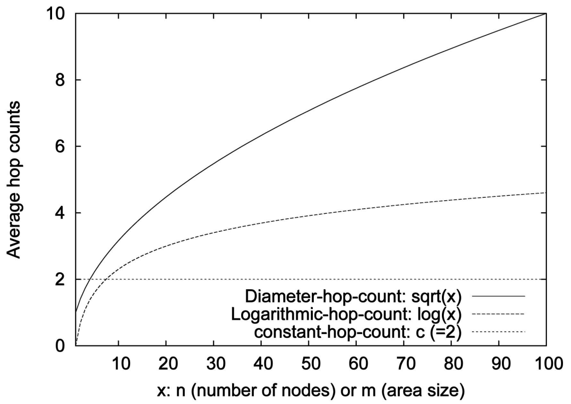


Fig. 1. Comparison of three different routing schemes in terms of average hop counts.

this approach is limited by the memory capacity of the ferry.

In this paper, we focus on the controlled movement-assisted model and propose a new model that avoids these two extremes in terms of hop count. A moderate constraint $\log m$ is given in an $m \times m$ square area, and the corresponding method is called logarithmic hop count (see Fig. 1).[1] This constraint is enforced by a hierarchical structure of trajectory for ferries. In a nutshell, we consider a new routing process in a likely unconnected manet by considering message routing, as well as trajectory planning, of ferries that carry the message. Nodes are stationary unless they are instructed to move (and become ferries) for message deliveries. For each routing, the number of relays is bounded by $\log m$, whereas the expected total moving distance of ferries for each communication is bounded by the $O(\log \frac{m}{d})$ factor of the physical distance d between the source and destination. Also, by enforcing a waiting time at each *rendezvous point* (that is, the place where a message forwarding occurs), each ferry can potentially carry multiple messages and thereby reduce the total number of ferries. Using replications increases the number of relays in the worst case.

The proposed scheme resembles several existing movement-assisted schemes, including MF with multiple ferries [7], with several key differences: 1) *Different objectives*. Although most existing movement-assisted schemes have competitive results on hop count, other metrics fare poorly, especially average delay. Our approach can achieve an average bound $O(\log \frac{m}{d})$ on moving distance with respect to the physical distance between the source and destination with logarithmic hop count. As a by-product, our approach automatically redistributes nodes to "hot" regions with high traffic volume thus achieving dynamic load balancing and improving the overall performance. 2) *New technical issues*. We address several new issues in a system where the role of each node (ferry or not) is dynamically determined based on the network configuration. These issues include onDemand ferry solicitation, dynamic trajectory planning of ferries, rendezvous point placement, and adaptive ferry migration and load balancing. The main contributions of this paper are the following:

1. We advocate the importance of message routing, as well as trajectory planning, of ferries that carry the message in a routing protocol and study the following three issues:
   - Propose a logarithmic store-carry-forward scheme through a hierarchical structure of trajectory for ferries that controls the number of relays.
   - Offer dynamic trajectory planning based on the network dynamic using either proactive or reactive methods of maintaining ferry movement.
   - Design for the final location of each ferry to balance node distribution in the sparse mode.
2. We conduct a comprehensive performance evaluation by giving analytical results on various metrics and conducting extensive simulations to compare the proposed scheme against the traditional and movement-assisted models in terms of network capacity (related to hop counts), moving distance, and delay.

The following assumptions are used in this paper: 1) The nodes are distributed in a given $m \times m$ square area partitioned into a mesh of 1×1 unit grids. The transmission range is set in such a way that any two nodes in a unit grid can communicate with each other. Each grid is associated with a geographical address. The communication is grid to grid. 2) Each node (including the source) knows its (grid) location and the (grid) location of the destination. This can be achieved through GPS or non-GPS localization methods [8]. In particular, when the communication is node to node, that is, the destination refers to a particular node (rather than a geographical location), some form of location management will be used, such as home region [9], to track the nodes. 3) Nodes in the systems are mobile (with either random or controlled movement). These nodes, called ferries, can be instructed to move for delivering a message. The ferry moves with constant velocity v. Message exchange between two nodes within each other's transmission range can be done instantly. In real applications, only a subset of deployed nodes needs to be mobile and controllable in terms of their movement.

The remainder of the paper is organized as follows: Section 2 discusses related work on mobility control in MANETs, with a focus on movement-assisted routing. Section 3 proposes logarithmic store-carry-forward routing (SCFR) in both dense and sparse modes of MANETs. Analytical results of various performance metrics are presented in Section 4. Some extensions to the basic scheme are presented in Section 5. Section 6 provides comprehensive simulation comparisons among different routing schemes, and conclusions are drawn in Section 7.

2 Related Work

Extensive work has been done on designing routing protocols in MANETs [10], [11]. These routing protocols are all based on the assumption that the network is

[1] When the network is sufficiently dense $m = O(\sqrt{n})$ and $\log m = O(\log n)$.

connected. In reality, the network could be highly partitioned due to the low density of the network and also the movement of nodes. These networks are known as delay-tolerant networks (DTNs) [12] (also disruption-tolerant networks [13]). Several models [14], [15] have been proposed to abstract DTNs in a graph model by incorporating time. More recently, a complete architecture to support various protocols in DTNs was designed [16]. DTNs include some other networks such as the underwater acoustic networks [17] and vehicle networks [18] with a set of random vehicles moving along a set of tracks.

The existing movement-assisted routing methods in DTNs can be classified into two categories according to the mobility control. The first category exploits the mobility of nodes to transmit messages but does not change their original random movement. The second category is controlled movement, where nodes may change their original trajectory to deliver messages.

ER [6] is the typical random movement scheme. It is a flooding-based algorithm, where nodes are all mobile and have infinite buffers. When a node has a message to send, it propagates the message to all nodes it meets, which continue to propagate the message. Eventually, the data is delivered to the destination with a high probability in a bounded amount of time. In [19], the ER method is extended by considering finite buffers. Therefore, a drop strategy is developed by exploiting node mobility statistics. Animal tracking networks such as SWIN [20] and ZebraNet [21] also use random node mobility and flooding-based relaying.

MF [5], [7] is one of the most important methods using controlled node movement. In MF, some ferries, which are nodes that have completely predictable routes through the geographic areas, are employed for message delivery. Nodes route messages end to end using the ferries. Ferries move around according to the known routes, typically a Hamiltonian circuit. Multiple ferries could be deployed in a set of subregions through partitioning, which leads to several MF extensions with no ferry interaction, mobile node (ferry) relaying, or static node relaying. The idea of using ferries has been applied in several rural connectivity projects for providing Internet access to remote areas [22], [23].

Limited work has been done using loosely controlled or a hybrid of random and controlled models. The work in [24] recruits mobile hosts as intermediate nodes that actively modify their trajectories to transmit messages. It aims at minimizing the modification of trajectories while getting the message across as fast as possible. The work in [25] focuses on the adjustment of node distribution on a fixed route for efficient message delivery. However, most existing work on trajectory control is limited to dealing with energy conservation. MV (meetings and visits) routing [26] is the only existing work that actively recruits mobile nodes to help relay, whose trajectory is controlled on demand to meet a certain criterion, but global route sharing is not considered.

There could be other classifications such as message control, which deals with how a routing message is handled during the routing process, including the copy method (single copy [27], [28], multiple copies [29], and flooding [6]), the number of relays, and memory capacity, and collective communication, whether the message delivery is unicast, anycast, multicast, or broadcast. Also, the connection-based and movement-assisted-based models could be combined to provide high-performance data delivery in some specific systems [18].

3 The Logarithmic SCFR

In this section, the basic idea of the proposed logarithmic SCFR method will be introduced first, followed by a detailed design for both dense and sparse modes of MANETs.

3.1 Basic Ideas

SCFR consists of four parts:

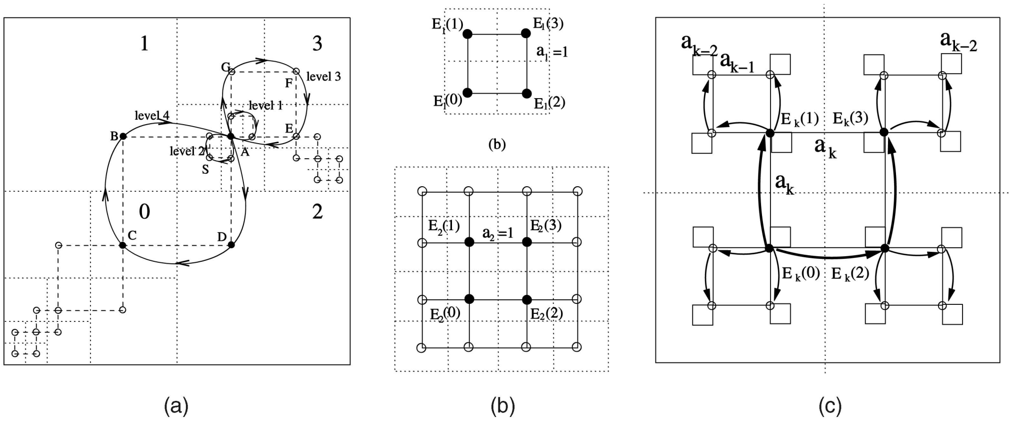
1. Design of trajectory and rendezvous points for ferry.
2. Movement scheduling of both ferry and message.
3. Dynamic role change of node, including on_demand ferry solicitation.
4. Node redistribution through ferry movement.

We propose hierarchical trajectories for ferries. Each ferry follows a trajectory that covers a certain region, which includes subregions covered by ferries at lower levels in the hierarchy. Rendezvous points are transfer stations located on the crossing of two trajectories of different levels. The ferry of one level will carry the message to the rendezvous point that is the closest to its destination and drop it there. The ferry of the other level will pick up the message there and deliver it to its next relay. To facilitate the routing process, we also need to translate the geographical location address to one that better reflects the subregion address in the hierarchy.

The movement scheduling of ferry and message in the hierarchical model is similar to a public transportation system. The interstate railway carries passengers to every state with the state capital as the station, and another intrastate railway delivers passengers to every city (including the state capital) in the state. Fig. 2a illustrates the general hierarchical trajectory structure in a $2^4 \times 2^4$ mesh. Ferries at the highest level, level 4, circle the largest square $ABCD$ with each corner being the rendezvous point, whereas ferries at level 3 circle $AEFG$. Fig. 2a also shows level 2 and level 1 ferry trajectories that go through the rendezvous point A.

At each rendezvous point, a node called a *keeper* is selected as the storage station. A keeper may serve as a storage station for several levels, as node A does for four levels in Fig. 2a. One or more nodes in each grid are solicited on demand as ferries and circle the trajectory loop to visit four keepers in the loop one by one. All other nodes that are neither keepers nor ferries can simply be called *ordinary nodes*. The roles of ferry and keeper can be changed, together with ordinary nodes.

SCFR can operate in two modes: *dense* and *sparse*. The dense mode corresponds to networks with sufficient nodes in each grid for keeper selection and ferry solicitation. In the sparse mode, a ferry may not find a keeper in a rendezvous point, and a keeper may not find a candidate for ferry solicitation. In these cases, the ferry and keeper have to change their roles and pick up the work of the missing keeper



(a) (b) (c)

Fig. 2. (a) Movement of ferries in a $2^4 \times 2^4$ mesh. Illustration of "eye" squares of (b) a 2×2 mesh, (c) a 4×4 mesh, and (d) a $2^k \times 2^k$ mesh and the optimal TCD when the source is an eye $(E_k(0))$.

and ferry. In a network with unbalanced node distribution, such role changes cause node migration and improve system-wide load balance. For example, in Fig. 2a, a ferry of level 2 can be changed to the one for level 4 at grid A to deliver messages from S to D. We can view the trajectory of a ferry in the dense mode as a trajectory of a public bus that is fixed, and the trajectory of a ferry in the sparse mode as a trajectory of a carpool taxi that is determined on demand by customers.

3.2 Hierarchical Trajectory

In the proposed hierarchical trajectory for ferries, each trajectory is a square loop with four corners (rendezvous points). Once the locations of the four rendezvous points (called *eyes* [30]) are defined, the trajectory of the corresponding level is formed. Basically, the given $2^k \times 2^k$ 2D mesh is partitioned into four subregions repeatedly k times, and k levels of eyes are generated from the highest level k to the lowest level 1.

Definition 1. *All four nodes in a 2×2 mesh are eyes of level 1 $E_1(j)$, $j = 0$, 1, 2, 3. A $2^i \times 2^i$ mesh is partitioned into four $2^{i-1} \times 2^{i-1}$ submeshes, each of which has four eyes $E_{i-1}(j)$. Eyes $E_i(j)$ are selected from 16 $E_{i-1}(j)$s, specifically, eyes $E_i(j)$ are the four $E_{i-1}(j)$s that are the closest to the center of the $2^i \times 2^i$ mesh (see Figs. 2b, 2c, and 2d).*

Define the square formed by the four eyes ($E_i(0)$, $E_i(1)$, $E_i(2)$, and $E_i(3)$) at level i $(1 \leq i \leq k)$ as its four corners to be the eye square of the $2^i \times 2^i$ submesh. Denote a_i as the length of the side of this eye square. As shown in [30], when there are k levels, $2a_k + 2a_{k-1} = 2^k$. For example, for $k = 5$, $a_1 = 1$, $a_2 = 1$, $a_3 = 3$, $a_4 = 5$, and $a_5 = 11$. In general, we have $a_i = \frac{1}{3}[2^i - (-1)^i]$.

The eye theory [30] was originally proposed to achieve an optimal total communication distance (TCD) algorithm for a time-optimal broadcast in a wormhole-routed 2D mesh without traffic contention. Suppose a TCD is the summation of all the distances a broadcast message traverses during the process, with the highest level eye being the source. A minimum TCD among TCDs for all the possible source nodes is guaranteed [30]. Fig. 2d shows a minimum TCD broadcast algorithm that starts at the highest level eye. The process starts from the highest level k to the lowest level 1. A total of $2k$ steps are needed with two steps in each level. If we view many pairs of unicasts, which could share ferries as broadcasts, and set the trajectory of each ferry to be the loop formed by the four eye positions of a certain level, our problem can be approximated to the eye theory and good performance with respect to a short moving distance that can be expected.

Before defining the routing process, we first need to translate the geographical location address to a special quaternary address that better reflects the subregion address in the hierarchy: When the area is divided into four quadrants, each quadrant gets one address. The bottom left one is "0," and the others get "1," "3," and "2" in the clockwise direction. This process repeats in each quadrant until the unit grid is reached. Thus, each unit grid has an address of length k.

Address Transfer Rule. *Given a coordinate of a grid (x, y), make x into a binary number $x_k x_{k-1} \ldots x_1$, y into $y_k y_{k-1} \ldots y_1$, and $x_i y_i$ into a quaternary q_i, for $i = 1, \ldots, k$. Then, $q_k q_{k-1} \ldots q_2 q_1$ is the special quaternary address for the grid.*

Fig. 3b is the corresponding quaternary address for the grids in Fig. 3a. The submesh that corresponds to quadrant 2 is shown in a bold square in Fig. 3b.

Eye Position Rule. *Given a unit grid with the quaternary address q, $q = q_k q_{k-1} \ldots q_2 q_1$, 1) it is an eye of level 1 and 2) if it is the eye of level i and $q_{i+1} + q_i = 3$, then it is an eye of level $i + 1$.*

Based on the eye position rule, we can easily decide whether a grid is the eye of level i using the special quaternary address. Fig. 2a is a $2^4 \times 2^4$ mesh. Thus, according to the address transfer rule, the special quaternary address for grid A is 3030. When the eye position rule is applied to A, since $3 + 0 = 0 + 3 = 3$, it is the eye of levels 1, 2, 3, and 4. Similarly, grid E (3212) is the eye of levels 1, 2, and 3 since $1 + 2 = 2 + 1 = 3$ but not of level 4 since $3 + 2 \neq 3$.

3.3 Movement Scheduling of Ferry and Message

As opposed to the traditional connection-based routing (CB), which schedules the "next hop" for each message, and the mobility-assisted (controlled) approach, which focuses on the trajectory design of ferries, our approach needs to

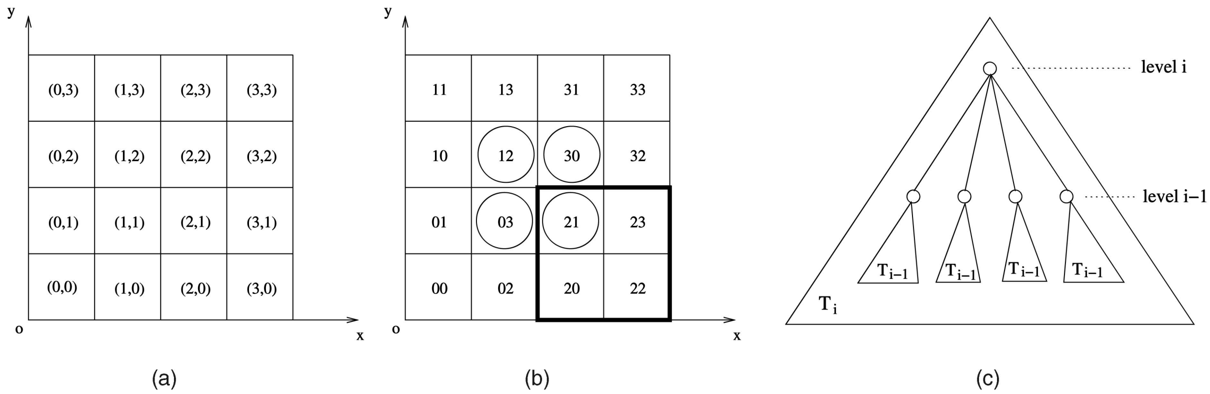


Fig. 3. Illustration of grid addresses. (a) The coordinate addressing. (b) The special quaternary addressing. (c) The quaternary tree.

design paths for both messages and ferries. This is because ferries of different levels follow their trajectories, whereas messages should be relayed several times by the ferries before reaching the destination. Like passengers in a transportation system, messages have to make a sequence of decisions before arriving at the final destination. They need to select each intermediate stop (keeper) and the vehicle (ferry) that will take them to these stops.

Each ferry of level i circles an eye square of level i in the clockwise direction. If we extract the $i\text{th}$ digit of the address of the four eyes at level i, the ferry follows the loop $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 0$. The ferry in a particular loop may leave its eye-square loop in the sparse mode as will be discussed later. Each relay in a routing process is determined via an *address-matching process*. To simplify the description, we view the hierarchical trajectory as a quaternary tree shown in Fig. 3c, where the root of T_i is connected to roots of four subtrees. However, this root is just a placeholder for one of four roots of the subtrees. Therefore, a grid may appear once at each level from level 1 to level i according to the eye position rule.

The routing process starts with finding a *least common ancestor* (LCA) of source and destination. In the first phase, the source sends the message to LCA by "moving up" the quaternary tree. In our address-matching process, suppose the highest unmatched digit between the source and the destination is p. The LCA corresponds to an eye of level p with the same prefix of source address. Once the highest unmatched digit is resolved through multiple levels of moving up, the second phase corresponds to a sequence of "moving down" the quaternary tree, with each level resolving one digit from left to right without altering the matched prefix. Moving up or down between adjacent levels can be done either by a ferry circulating among the four eye grids or with no action needed when two placeholders correspond to the same grid.

In Fig. 4a, suppose the source S is 0001 (meaning S in quarter 1 at level 1 and in quarter 0 at all other levels), and destination D is 3222. When performing the matching, the first unmatched digit is 4; thus, the next hop of this message should be the eye of level 4 with the address beginning with 3. Thus, according to the eye position rule, the next hop address is 3030 (LCA of S and D), position A. To reach A at level 4, the message may be relayed several times by carriers of different levels, from level 1 up to level 4. When the message reaches A via track transfers at G (0003), F (0030), and E (0303), it repeats the above matching process in a quarter region by resolving the remaining unmatched digits. The first unmatched digit of 3030 and 3222 is 3; thus, its next hop is the eye of level 3 with 32 as the address

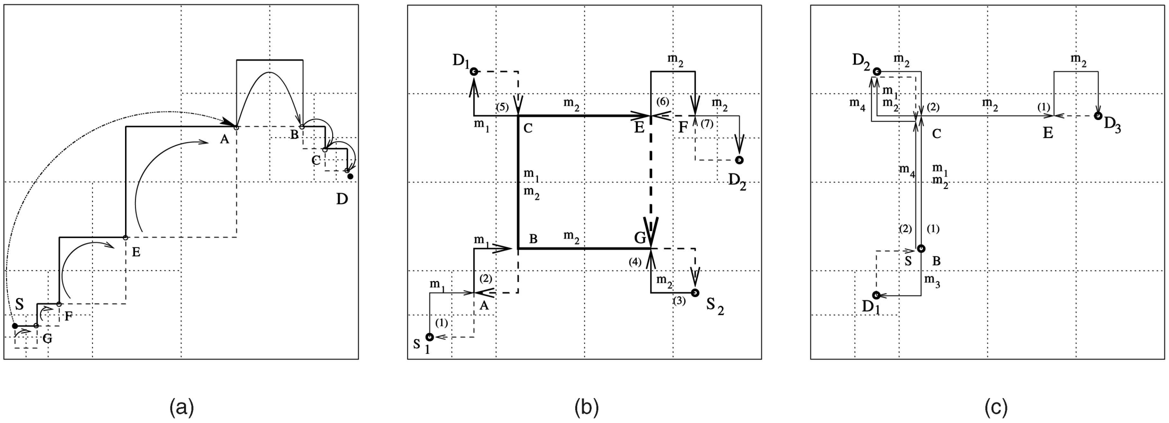


Fig. 4. (a) Ferry movement schedule. A possible message routing path from S to D $(k = 4)$. (b) Ferry solicitation and (c) ferry and keeper redistribution are two examples.

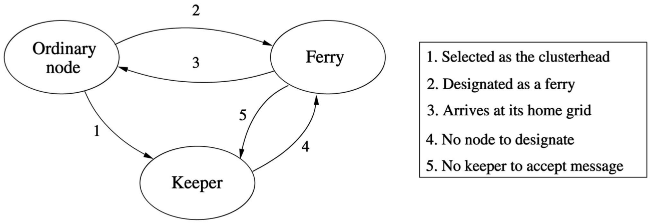

Ordinary node
Ferry
Keeper
1
2
3
4
5
1. Selected as the clusterhead
2. Designated as a ferry
3. Arrives at its home grid
4. No node to designate
5. No keeper to accept message

Fig. 5. Node status change in SCFR.

prefix. It is node B (3212). The next hop then is C (3221), and the final hop is D (3222). Note that selecting an appropriate branch of the four subtrees comes naturally, since the ferry will visit the root of the each subtree in sequence.

The routing process is implemented in a localized way by comparing the current $(curr)$ and destination $(dest)$ addresses to determine the "next hop"—a level transfer at a rendezvous point (that is, eye). Since a grid may appear in multiple levels, moving up and down the quaternary tree can be accelerated by skipping levels. We assume that h is the highest eye level of $curr$ using the eye position rule. We denote the first unmatched digit of $curr$ and $dest$ as p, and the next hop of the message is at level $\min(h, p)$. We can view the skipping level mechanism at a grid as an "express lift" that moves up and down the quaternary tree without forwarding cost or delay.

For example, in Fig. 2a, we assume the source is S (3003) and destination is C (0303). When we first compare S with C, the first unmatched digit is 4. However, since the highest eye level of S is 2, the next relay of this message is at level 2 (instead of level 1). Thus, the message will be picked up by a ferry that circles level 2 and later dropped to grid A (3030). Then, the first unmatched digit of A and C is still 4, and since A has the highest eye level of 4, the message will be picked up by a ferry circling level 4, skipping the journey of circling level 3. In this example, LCA is C with no moving down phase. If we exchange the role of S and D, LCA is still C with no moving up phase. Again, level 3 is skipped in the moving down phase.

3.4 Dynamic Role Change of Node

Each node is in one of the three states: ferry, keeper, and ordinary node, as shown in Fig. 5. Ferries are moving; keepers and ordinary nodes are stationary. A keeper is elected in every nonempty grid. Ordinary nodes generate messages and pass them to local keepers. When a keeper has a message to a remote grid, it solicits an ordinary node as a ferry to carry the message. After the message arrives at the next eye grid, it continues its journey with a new ferry solicited by the next keeper. This process repeats until a ferry arrives at the destination. For simplicity, we first consider the dense mode: Each intermediate eye of a route is nonempty such that a keeper is available to receive the message. In addition, whenever a keeper needs to solicit a ferry, there is always an ordinary node in the same grid.

To control the number of ferries, a keeper does not solicit a ferry immediately after receiving a message. Instead, it starts a timer of t_i and expects a level i ferry to pass by during this period. A new ferry is solicited only after the back-off timer expires. The timer is set to the time that it takes for a ferry to travel one side of a level i eye square, that is, $t_i = a_i/v$. This prevents the number of ferries from exceeding four per eye square. Since a keeper may be responsible for several levels, it may use up to k timers, one for each level. As ferries are dynamically solicited, they are dismissed when there is no message to carry. In this case, a ferry returns to its original grid and becomes an ordinary node. For this purpose, each node keeps a variable $home$, which records its last position when it is solicited.

Fig. 4b shows an example of SCFR in the dense mode. Node S_1 is sending a message m_1 to D_1, and S_2 is sending m_2 to D_2. Message m_1 is carried to the top-level eye B by ferries 1 and 2. Meanwhile, message m_2 arrives at another top-level eye G, which solicits ferry 4 to carry this message. Ferry 4 arrives at B before timer expiration and carries both m_1 and m_2. Message m_1 is dropped at the next eye C, where it is picked up by ferry 5 and delivered to D_1. Message m_2 travels alone; it passes eyes E and F before it reaches D_2. A total of seven ferries are involved in the routing process. If there are no more messages to deliver, all these ferries will return to their home grids (represented by dashed arrows) and switch to ordinary nodes.

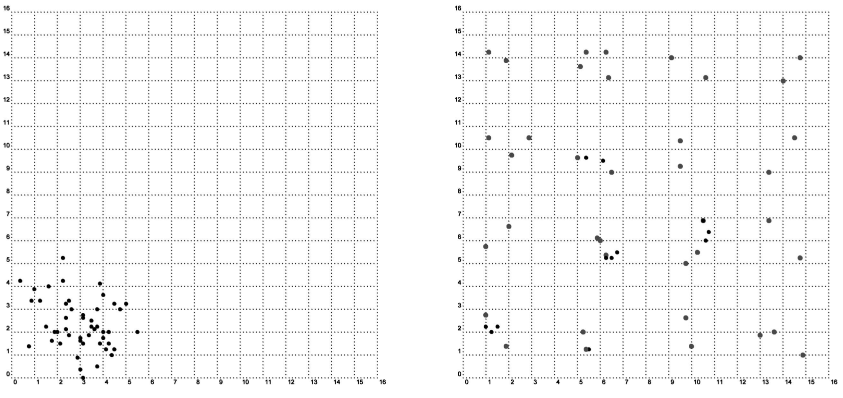
3.5 Adaptive Node Redistribution

In the sparse mode, some intermediate keepers may fail to solicit ferries and some intermediate eyes are empty and, thus, have no keepers. As a solution, we propose a ferry and keeper redistribution scheme, which solicits ferries and keepers from dense and low traffic regions and redistributes them to regions with high data traffic. Whenever a ferry F enters a grid without a keeper, it stays and acts as a new keeper K that has received all the messages from F. When a keeper K has an expired timer, but cannot find an ordinary node in its grid, it solicits itself. That is, it acts as a new ferry F' to carry all messages in K. Imagine an extreme case with only two nodes in the network. The source node will travel all the way to deliver the data to the destination. Its status will alternate between the keeper and the ferry at each intermediate node.

Ferries in SCFR are stateless. They "lose memory" after being converted to a keeper. When the converted keeper becomes a ferry again, its $home$ grid is set to its current position instead of its original grid. This mechanism allows ferries to migrate to regions with needs and optimizes the node distribution over time. When there are multiple messages waiting, a keeper may have several ticking timers t_i for different level i, $1 \leq i \leq k$. In the dense mode, the keeper solicits a level i ferry when a t_i timer expires. However, when there is no ordinary node to solicit, a special rule is used.

Self-Solicitation Rule. *A keeper only solicits itself as a ferry that belongs to the lowest level of all levels with active timers.*

The above rule enforces a closest-destination-first policy. When one of the timers at a keeper expires, the keeper checks whether this timer belongs to the lowest level among all the active timers. If not, the keeper does not solicit itself until the lowest timer expires. We can view the storage space of a keeper for messages as a min-heap, with the level of each message as the sorting key. Every newly arrived message is inserted into the heap. The keeper solicits itself



(a) (b)

Fig. 6. A sample node redistribution of SCFR. (a) The initial node distribution. (b) The final node distribution.

as a ferry only when the timer of the top message in the heap expires. When messages are ordered by distances to their destinations and delivered in this order, the total delay is minimized. The detailed ferry and keeper behaviors are described in Algorithms 1 and 2.

Algorithm 1. Ferry $(i, curr, home)$ of SCFR

```
1. Updates its $dir$, moves to the next rendezvous point of
   level $i$, picks up any message.
2. When it has messages to drop at the rendezvous point of
   level $i$ based on the address-matching process, if there
   is a keeper, it drops the messages to the keeper;
   otherwise, it becomes the keeper of $curr$ and stops.
3. When $curr = home$ and it has no messages to send,
   it stops and becomes an ordinary node, or a keeper if
   there is no keeper; otherwise, it repeats step 1.
```

Algorithm 2. Keeper $(curr)$ of SCFR

```
1. For each new message, it finds the next hop at level $i$ for
   it, based on the address-matching process. If there is no
   timer for level $i$, it sets $t_i$.
2. When it catches a by-passing ferry of level $i$, it loads
   messages of level $i$ to it if any, and cancels the timer $t_i$.
3. When a timer $t_i$ expires,
   a. if there are ordinary nodes in this grid, it designates
      an ordinary node to be the ferry of level $i$, loads the
      messages for $i$ to it, sets its $home$ to $curr$, and cancels
      $t_i$;
   b. if there is no ordinary node in this grid, and $i$ is the
      lowest level among all active timers, it becomes the
      ferry of level $i$ with the $home$ set to $curr$.
```

Fig. 4c demonstrates the node redistribution process in SCFR. Node S is sending three messages: m_1 and m_4 to D_2, m_3 to D_1, and m_2 to D_3. Nodes are densely deployed in eye B, but the other eye grids (C and E) are empty. Assume messages m_1, m_2, m_3, and m_4 were generated in order, and m_3 was generated just before the timer of m_1 expires. According to the self-solicitation rule, the keeper in grid B waits until the timer of m_3 expires, which is of the lowest level, then it solicits ferry 1 to carry messages m_1, m_2, and m_3 to D_1 and drops m_3. Then, ferry 1 returns to S, and after a waiting time with role changes, it takes off to C. As the next eye grid C is empty, ferry 1 becomes its keeper and waits for a level 2 (for m_1) and a level 3 ferry (for m_2). After the level 2 timer expires, it solicits itself and delivers m_1 to D_2. Then, it travels to D_3, alternating its role between a keeper and a ferry. Finally, it stops at eye E and becomes a keeper. Message m_4 that is available at a later time is delivered by ferry 2, also solicited from grid B to D via C. After delivery, ferry 2 becomes a keeper of eye grid C. After the node redistribution, faster delivery is expected to destinations D_2 and D_3 since C has its own keeper.

A larger example of adaptive node redistribution can be found in Fig. 6. Initially, most nodes are deployed at the bottom left corner, which occupies about 10 percent of the deployment region. After 100 messages, these nodes tend to be evenly distributed to the entire region.

4 Properties

In this section, we analyze several performance metrics of SCFR. Since MF [7] also uses controlled ferries to help deliver messages, which is the closest method to our method, we use it as a counterpart for comparison in the analysis and also the simulation. The metrics we examine are listed as follows:

1. average/worst-case message moving distance (total traveling distance from source to destination) and average/worst-case distance stretch ratio,
2. average/worst-case delay (total time consumption from source to destination) and average/worst-case delay stretch ratio,
3. average/worst-case number of relays, and
4. number of ferries in the system.

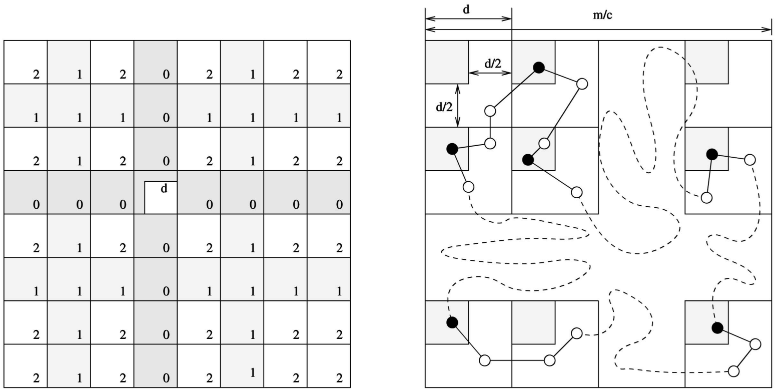
We assume a nonuniform keeper timer $t_i = a_i/v$, and later extend the case to uniform $t_i = 1/v$.

Lemma 1. *The message moving distance between two nodes, whose smallest common level is* i, *is* $d_i \in [\frac{1}{3}(2^i - 1), 3(2^i - 1)]$.

Proof. In the best case, the source and destination are two neighboring eyes of level i, and the moving distance is

$$d_i^{Min} = a_i \geq \frac{1}{3}(2^i - 1),$$

where a_i is the side length of a level i eye square, as defined earlier. In the worst case, the source and



(a) (b)

Fig. 7. (a) Worst-case and average stretch ratios $(l = 3)$. (b) Expected length of a Hamilton circuit in each cell.

destination are in different submeshes of the level i division, as shown in Fig. 4a, and in each eye square, the data moves three fourths of the perimeter. The total moving distance is

$$d_i^{Max} = 2\sum_{j=1}^{i-1}(3a_j) + (3a_i) = 3(2^i - 1).$$

□

Theorem 1. *SCFR has the worst-case message moving distance of $O(m)$ and the average of $\Omega(m)$.*

Proof. The worst case distance is a special case of Lemma 1, that is,

$$d^{Max} = d_k^{Max} = 3(2^k - 1) = 3(m - 1).$$

To calculate the average case distance, we assume a uniform distribution of the source and destination nodes. The probability of a smallest common level i is $(\frac{1}{4})^{k-i} \times \frac{3}{4}$. Therefore, the average moving distance is

$$d^{Avg} = \sum_{i=1}^{k}\left(\frac{1}{4}\right)^i\left(\frac{3}{4}\right)d_i.$$

From Lemma 1, we have $d^{Avg} \in [\frac{2}{7}m, \frac{18}{7}m]$. □

Consider two nodes with distance d and the corresponding message moving distance d'. We define the distance stretch ratio as $\gamma = d'/d$. For simplicity, we measure d as $\max(d_x, d_y)$, where d_x and d_y are the x and y offsets in the number of grids, respectively.

Theorem 2. *SCFR has the worst case distance stretch ratio $O(m/d)$ and the average of $O(\log(m/d))$.*

Proof. Without loss of generality, we assume the source is on the top and left side of the destination. The worst case happens when the two nodes have the smallest common level of k (the source lies in the central $d \times d$ white square in Fig. 7a). The corresponding stretch ratio is

$$\gamma^{Max} \leq d^{Max}/d = 3(m - 1)/d.$$

Then, we show that the average stretch ratio is logarithmic. Letting $l = \lfloor\log(m/d)\rfloor$, we divide the $m \times m$ area into 4^l squares with side length $2^{\lceil\log d\rceil}$, as shown in Fig. 7. These squares have types $0, 1, \ldots, l - 1$. When the source is in a type i square, the smallest common level of the two nodes is at most $k - i$, and the corresponding stretch ratio is at most

$$\gamma_i^{Max} \leq d_i^{Max}/d < 3(2^{k-i})/d = 3(m/d)2^{-i} < 3(2^{l-i+1}).$$

The probability that the source is in a type i square is less than $2(2^i)(2^l)/4^l = 2^{i+1-l}$. The expected stretch ratio is

$$E[\gamma] < \sum_{i=0}^{l-1}\gamma_i^{Max} \times 2^{i+1-l} = \sum_{i=0}^{l-1}3(2^{l+1-i})(2^{1-l}) = 3(2^2)l$$
$$< 12\log(m/d).$$

□

Theorem 3. *SCFR has the worst case delay of $O(m)$ and the average of $O(m)$.*

Proof. In SCFR, the latency is the sum of the time for moving and the time for waiting. The first part is proportional to the moving distance, which is $O(m)$ in both worst and average cases (Theorem 1). The total waiting time is

$$W^{Max} = t_k + t_{k-1} + \ldots + t_1 + t_2 + \ldots + t_k$$
$$= d^{Max}/\frac{3}{4} = O(m).$$

The average waiting time $E[W]$ is also bounded by $O(m)$. The overall delay is $O(m)$ in both worst and average cases. □

Similar to the distance stretch ratio, we define the delay stretch ratio as the ratio between the actual delay to the minimal delay using a single ferry via a shortest trajectory. The following result can be proved in a similar way: SCFR has the worst-case delay stretch ratio $O(m/d)$ and the average of $O(\log(m/d))$.

Theorem 4. *SCFR has the worst case forwarding number $O(\log m)$ and the average of $O(\log m)$.*

Proof. In SCFR, there are k levels of ferry loops, and when messages are transferred between two levels, it is forwarded twice. One is from the ferry of level i to the keeper, and the other is from the keeper to ferry of level $i-1$ or $i+1$. In the worst case, when the source and destination are in different submeshes of the first-level division, messages will be relayed $2(2k-1)$ times, and $F^{Max} = 2(2k-1) = O(\log m)$. In the average case, we use the same method as the proof of Theorem 1:

$$E[F] = \sum_{i=1}^{k} \left(\frac{1}{4}\right)^{k-i} \left(\frac{3}{4}\right) 2(2i-1) = O(k) = O(\log m).$$

□

Theorem 5. *SCFR has at most $O(m^2)$ ferries.*

Proof. In SCFR, there are at most four ferries in each eye square. The number of eye squares at level i is 4^{k-i}. The maximal number of ferries is

$$\sum_{i=1}^{k} 4(4)^{k-i} = 4(4^k/3) = \frac{4}{3}(2^k)^2 = O(m^2).$$

□

Among the four schemes of MF in [7], we focus on the node relaying algorithm (NRA), which divides the deployment area into $c_1 \times c_2$ cells (we make $c_1 = c_2 = c$ here for fair comparison, where c is a constant). A ferry is assigned to each cell, and intercell transfer is conducted via a node that belongs to both cells to relay data. Here, this node can be viewed as the transfer station, and no synchronization is needed. Within each cell, the ferry travels along a loop that passes through every node and has the smallest perimeter (the Hamiltonian circuit). The solution for the traveling salesman problem can be used for it.

Theorem 6. *MF has the worst-case message moving distance of $O(m\sqrt{n})$ and an average of $O(m\sqrt{n})$.*

Proof. We first consider the average moving distance within each cell. As shown in Fig. 7b, a square cell of side m/c can be divided into n/c^2 grids of side $d = m/\sqrt{n}$. Consider the gray square of size $d/2$ in the upper left corner of each grid. Let X_i be the indicator variable for a nonempty gray square:

$$Pr(X_i = 1) = 1 - \left(1 - \frac{1}{4n}\right)^n > \frac{1}{8}.$$

Let X denote the number of nonempty gray squares in a cell, we have

$$E[X] = \sum_{i=1}^{n/c^2} X_i = \frac{n}{8c^2}.$$

From each nonempty square, select a node x_j (the black node) such that their sequence in the Hamiltonian circuit is $v_1, v_2, \ldots, v_X$. The length of each Hamiltonian circuit segment from v_i to v_{i+1} $(1 \le i < X)$ and that from v_X to v_i is at least $d/2$. Then, the expected length of the ferry trajectory in each cell P_c is

$$E[P_c] \le E[X] \times \frac{d}{2} = \frac{m\sqrt{n}}{16c^2}.$$

In both the worst and average cases, data travels $O(c)$ cells and an $O(m\sqrt{n}/c^2)$ distance in each cell. The corresponding total distance is $O(m\sqrt{n}/c)$. □

Since there is no waiting time in MF, the average (or worst) case delay is still $O(m\sqrt{n})$. The number of relays is $O(1)$, and the number of ferries is $O(1)$.

5 Extensions

In this section, some extensions of the proposed SCFR are provided, which aim at balancing the workload to further improve the system performance.

5.1 SCFR with Uniform Backoff Time

In SCFR, the timer is set to the time that the ferry of that level travels one side of its eye-square loop, which is different in each level. Therefore, the maximum number of ferries at any time in a loop is four, which is consistent for loops at each level. Generally, as the traffic backbone, the higher level loops have more traffic than the lower ones. Instead of different timers for different levels, keepers can set the same timer for all messages. Thus, higher levels may contain more ferries to balance the load of ferries across the levels.

We can set the timer to be t_k, the time the ferry travels one side of the smallest loop, which is $1/v$. Also, the side length of an eye square at different levels a_i is known. Therefore, the delay in SCFR is shorter. The number of ferries in the system is larger when there is heavy traffic. The maximum number of ferries is thus $\sum_{i=1}^{k} 2^{2(k-i+1)} \times a_i = O(m^2)$.

5.2 Extensions to Ferries with Memory Capacity Limit

So far, we assume that each node can carry as many messages as needed. Obviously, a fixed amount of ferries in each eye square (loop) will result in different loads for ferries at different levels. The ferries at higher levels will carry greater loads than those at lower levels.

Again, assume the traffic is random but uniformly distributed. The amount of traffic that goes through a particular loop can be calculated by counting the different numbers of source and destination pairs that would go through the loop. Consider the main loop at level i in a $2^i \times 2^i$ submesh and let $d(i) = 2^i$ and again $m = 2^k$. We have the following result.

Theorem 7. *The number of routing pairs (source and destination) that would go through a level i loop is*

$$\rho(i) = 2d^2(i)\left(m^2 - \frac{5}{8}d^2(i)\right).$$

Proof. Consider the following cases: 1) The loop will be used when a pair is selected with one node in the $d(i) \times d(i)$ submesh and one outside the submesh. Therefore, the number of pairs is $2(m^2 - d^2(i))d^2(i)$. 2) The loop will be used when both nodes are from the same submesh. In this case, the submesh is partitioned into four $d(i)/2 \times d(i)/2$ submeshes and these two nodes come from two different $d(i)/2 \times d(i)/2$ submeshes. The number of pairs in this case is $4(d^2(i) - (\frac{d(i)}{2})^2)(\frac{d(i)}{2})^2$. The result is the sum of these two cases. □

When the traffic at a loop is defined as the number of routing pairs that go through it, let $\gamma(i) = \rho(i)/\rho(i-1)$ be the ratio of the traffic of a loop at level i to the traffic of a loop at level $i-1$. We can easily get: $\gamma(i) < 4$. In fact, $\gamma(i)$ is a monotonically increasing sequence. $\gamma(k) = 1.78$, $\gamma(k-1) = 3.51$, $\gamma(k-2) = 3.88$, and $\gamma(k-3) = 3.98$. Therefore, γ quickly converges to 4.

Basically, under the requirement that each ferry takes the same average load, $\gamma(i)$ defines the relative number of ferries between two loops at adjacent levels. Since the size of each loop is different, we need to respect the following equation when converting to the timer $\gamma(i)$ at each rendezvous point: $\frac{t_i}{t_{i-1}} = \frac{a_{i-1}}{a_i} \times \gamma(i)$. A simple analysis shows that when k is relatively large and i is relatively small, a_{i-1}/a_i is close to 1/2, the ratio of t_i/t_{i-1} is between 1 and 2.

5.3 SCFR as an Extended MF

In the dense mode, the SCFR can also be designed to work in a proactive mode. In proactive store-carry-forward routing (P-SCFR), there is a ferry circling each eye square, whether there is a message or not. One of the keepers on the eye square is responsible for designating the ferry.

After the setup procedure of the system, the ferries should be generated. The algorithms for the ferry and keeper in P-SCFR are quite simple. The ferry just circles its loop, collecting and delivering messages. It does not have a $home$. For a keeper, it finds the next hop for each message and waits for the corresponding ferry to send them. We can also employ two ferries in every eye square with each circling in a different direction to reduce the message moving distance and delay. However, the number of ferries is doubled. This proactive SCFR can be viewed as an extension for the work in MF. The difference is that a hierarchial structure is used here to reduce the average delay.

6 Simulation

In this section, we present the results of our simulation of the proposed movement-assisted routing algorithm SCFR in comparison with MF [5], [7] and ER [6]. Traditional CB is also used as a benchmark in the simulation.

6.1 Simulation Environment

All approaches are simulated on a custom simulator. We set up the simulation in a 128×128 square area, which is the target field. Nodes can be deployed in this area following certain distributions. We use uniform random distributions in the simulation except in one case, where we use a normal distribution where 90 percent nodes are in 25 percent region of the area, to examine the performance of SCFR in balancing node distribution. The parameters in our simulation are the following:

1. *Number of deployed nodes* n. We vary it from 20 to 1,000 to test the scalability of the system.
2. *Level number* k. We use 4 as the value of k in SCFR in most of the simulation. The side length of the unit grid is 8. Then, we analyze the performance change when the level number is 3 or 5, with side lengths of the unit grid being 16 and 4.
3. *Round number* r. We use it as the unit time. In each round, a ferry moves a unit distance; thus, the velocity is 8.
4. *Data generating probability* p. In each round, each static node generates a message with the probability p. This parameter controls the message ratio in the system.
5. *Different timer schemes*. We use the nonuniform timer scheme for most of the simulation. We then compare it with the uniform timer scheme (U-SCFR) to examine the effect.

The performance metrics are

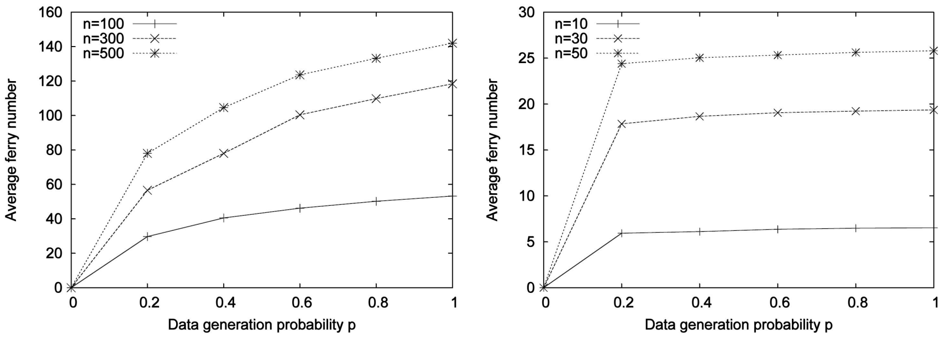
1. average message moving distance,
2. average message delay in terms of round, which is the sum of the traveling time and waiting time,
3. average number of relays for each message (or hop counts), which is related to network capacity, and
4. balance degree of final node distribution.

The last one is the side effect of the proposed algorithm, which tends to evenly distribute nodes. This in turn accelerates the message delivery procedure, hence improves the other metrics. The balance degree of node distribution is measured by the *standard deviation* of node numbers in all unit grids, which is the summation of the square of the difference between the load of each grid and the global average load, divided by the total number of grids and the *number of empty unit grids*.

When we implement the approach in MF and ER, we modify the algorithms for fair comparison. Since, in our approach we assume nodes in the same unit grid can communicate directly, we also apply the unit grid method to MF and ER. In MF, all the clusterheads in the unit grids are selected to form a ferry traveling circle. The source and destination are static nodes in SCFR and MF. Thus, in ER, we assume that each unit grid has a static node to generate and accept messages, and other mobile nodes pick up and drop messages when they meet with them. Since the transmission range only needs to guarantee the direct communication of nodes within a unit grid, which is quite small, only when the node number is large $(n > 100)$, the network is connected and CB can be applied. Thus, we use node number from 20 to 100 as the sparse mode and from 200 to 1,000 as the dense mode. In the simulation, we use the hop count of the shortest path between two nodes as their number of relays in CB. Except for when the round number is a tunable parameter, all the simulations conduct 10,000 rounds and results are gathered within this period. For each tunable parameter, the simulation is repeated 100 times or until the confidence interval is sufficiently small (±1 percent, for the confidence level of 90 percent).

6.2 Simulation Results

Fig. 8 shows the variation of the number of ferries in the SCFR system. Since SCFR is adaptive, the number of ferries increases with the traffic rate in the system, which is represented by the message generation probability, and also the number of total deployed nodes. The number of ferries increases with the amount of traffic, as well as the number of nodes. Fig. 8a is in the dense mode, and Fig. 8b is in the



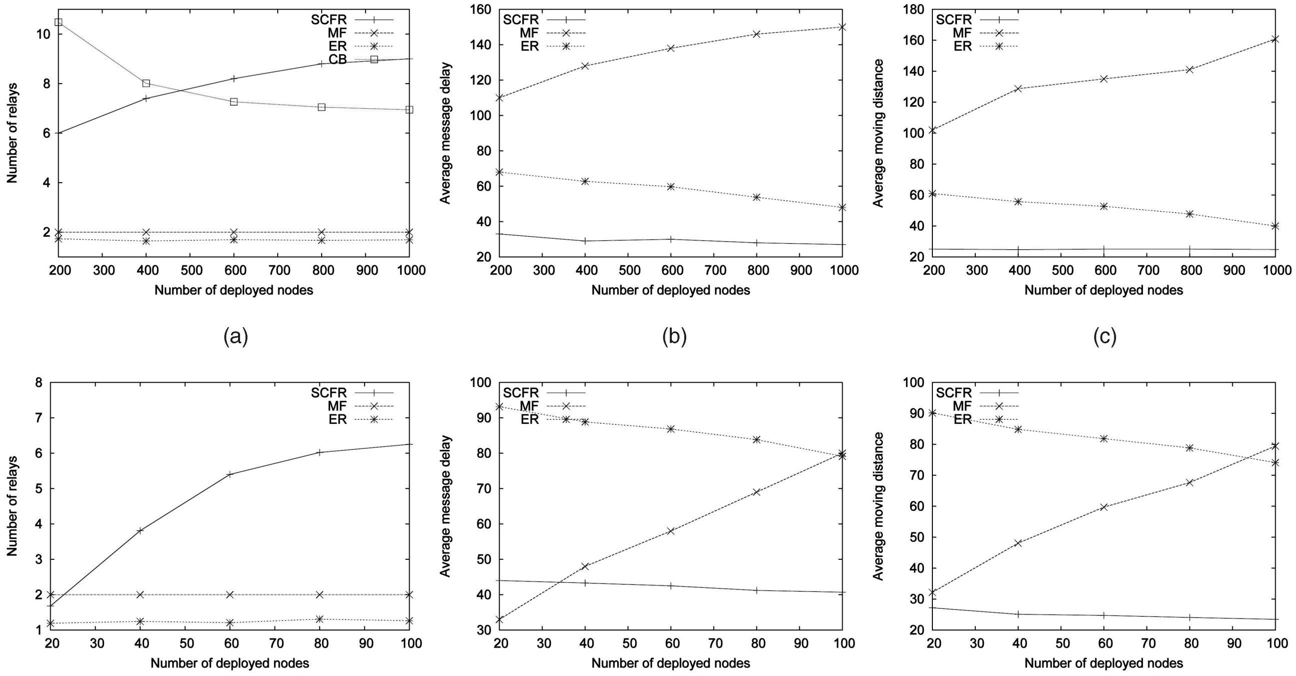
(a)

(b)

Fig. 8. Ferry numbers in SCFR with different p and n. (a) Dense mode. (b) Sparse mode.

sparse mode. We can see that with the growth of the number of nodes, the percentage of ferries decreases. The maximum percentage is around 50 percent. In the following simulation, we use a fixed value of 0.4 for p unless otherwise specified. MF and ER, which have a fixed number of mobile nodes, are not adaptive. For fair comparisons, we set the number of ferries in MF and the total number of mobile nodes in ER to 50 percent of the deployed nodes in the sparse mode and 25 percent in the dense mode, which are the average numbers of ferries in SCFR with p being 0.4.

Figs. 9a, 9b, and 9c show the performance comparison of SCFR, MF, ER, and CB in the dense mode. Figs. 9d, 9e, and 9f show the performance comparison of SCFR, MF, and ER in the sparse mode. Figs. 9a and 9d are the comparison in the number of relays. Although CB is superior in terms of latency and energy consumption, it has a large number of relays. Thus, we include the performance of CB in the dense mode as a comparison baseline. We can see that MF has a small fixed value of 2, which leads to high network capacity. ER also has a small value less than 2, which hardly changes with different numbers of nodes. This is because, in ER, although the message may be relayed many times to spread to the entire network, the copy with fewer relays tends to reach the destination with high probability since it exists for a longer time. Henceforth, it visits more area. CB has a large number of relays, especially when the network is relatively sparse. The number of relays in SCFR is determined by the number of levels k, the number of nodes in the system, and also the distribution of these



(a)

(b)

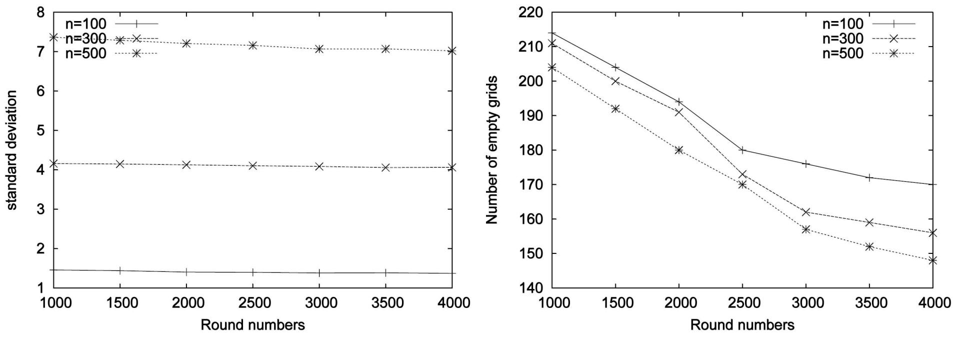
(c)

(d)

(e)

(f)

Fig. 9. Comparisons of SCFR, MF, ER, and CB in (a), (b), and (c) dense mode and (d), (e), and (f) sparse mode. (a) and (d) Number of relays. (b) and (e) Message delay. (c) and (f) Moving distance.



(a)

(b)

Fig. 10. Distribution variation with time (round number). (a) Normal deviation. (b) Number of empty grids.

nodes. When the network is sufficiently dense, it is a fixed value. When the network is sparse or the distribution is uneven, the number of relays decreases. This is because, in those cases, the ferry continues its role as a ferry, but enters other trajectories when it finds no keeper to drop the message and the supposed relay at the grid no longer exists.

Figs. 9b and 9e are the comparison in message delay. SCFR has the smallest value, and it changes slightly with the number of nodes especially in the dense mode. The message delay of MF increases with the number of nodes. When the number of nodes is sufficiently small, MF has smaller delay than SCFR. This is because with few nodes, the ferry delivery trajectory in MF is short, whereas in SCFR, the ferry has to make many detours to deliver the message with waiting time at each role change. The message delay in ER decreases with the growth of the number of nodes. This is because more mobile nodes carrying more copies help with faster delivery. ER has better performance than MF in the dense mode but worse in sparse mode.

Figs. 9c and 9f are message moving distance comparisons. These figures are similar to Figs. 9b and 9e. This is because delay is the sum of waiting time and traveling time, which can be represented by moving distance. In ER and MF, each message only has an initial wait to be picked up. Thus, the moving distance is slightly smaller than the message delay (we take the length of unit grid as length unit). In MF, an increase of the number of ferries may reduce the message delay, but its lower bound will be the traveling time. SCFR has the smallest moving distance, and in the dense mode, since the trajectory is fixed, the average moving distance is also fixed. In the sparse mode, it increases with the decrease of the number of nodes. The overall performance of SCFR is better than MF and ER with less message delay and moving distance and a larger, but still under control, (logarithmic) number of relays. MF has the second performance in practical, since the simulated ER algorithm assumes to have infinite buffer. The large number of copies of the message in ER, even with the TTL control, makes it impractical, whereas in both SCFR and MF, one copy for each message is kept.

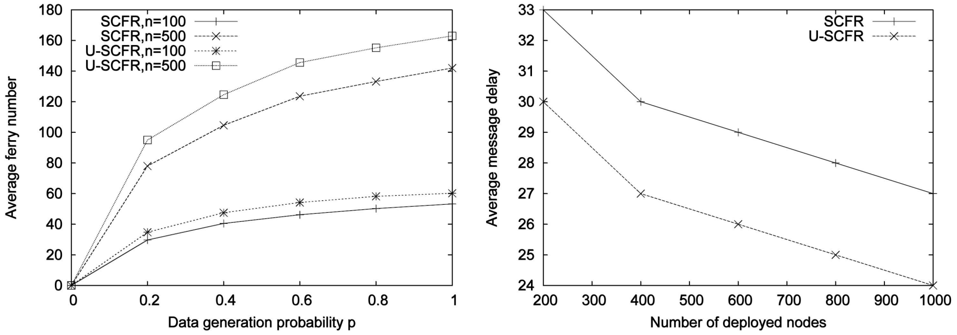
Figs. 10a and 10b show the effect of the redistribution of SCFR when the initial node placement follows a normal distribution. Fig. 10a is the normal deviation of the number of nodes in every grid, and Fig. 10b is the number of empty grids. With more rounds, the distribution tends to become more even. The number of empty grids decreases relatively fast, whereas the reduction of the normal deviation is not significant. This is because, although grids with more nodes generate ferries to fill the sparse region, the ferries terminate at eye positions, which leads to another kind of clustering. However, this kind of distribution benefits the data delivery system, since eye positions are the traffic backbone.

Figs. 11a and 11b show the performance comparison of SCFR (with nonuniform timer) with U-SCFR. Fig. 11a is the average number of ferries in the network. U-SCFR has more ferries, especially when the number of deployed nodes is large. This is because in U-SCFR, the timers are smaller than those of SCFR, except timers of level 1. Thus, more ferries tend to be employed. Fig. 11b shows the comparison of the average message delay. U-SCFR has smaller message delay than SCFR due to the smaller average waiting time. The delay decreases with the growth of the number of deployed nodes because more ferries are available.

Figs. 12a, 12b, and 12c show the performance comparison of SCFR with different parameter k (3, 4, or 5). Fig. 12a is the comparison of the average number of ferries in the network under different message rates. We can see that more levels lead to more ferries in the system, especially when the deployed nodes are plenty and the traffic rate is high. Fig. 12b shows the number of relays. More levels produce a larger number of relays. Fig. 12c is the comparison of average message delay. When k is 5, the delay is the largest. This is because more levels mean the message delivery trajectory is longer and the chances of waiting are higher. However, when k is small, the transmission range needs to be larger to make the direct communication available. When k is larger, the unit grid is smaller; thus, the balance effect of SCFR is more significant.

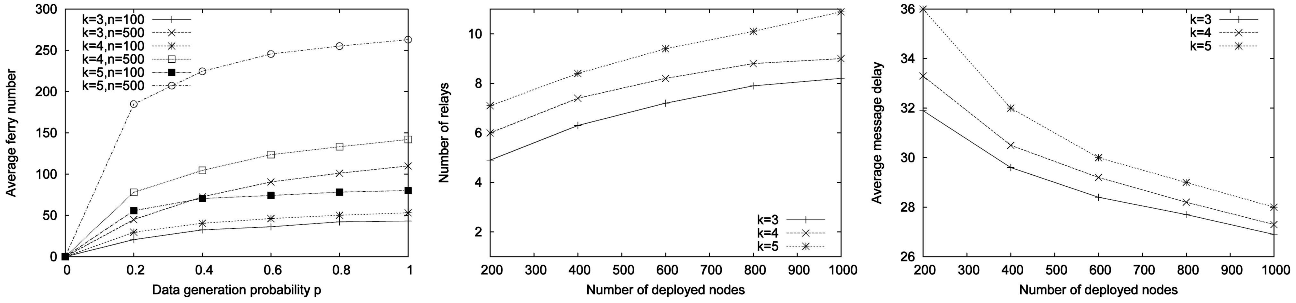
Simulation results can be summarized as follows:

1. SCFR is adaptive. When the message generating ratio is high, more ferries are generated to balance the workload of ferries.
2. SCFR achieves better performance in message moving distance and message delay than MF and ER, especially in the dense network.
3. SCFR has lower network capacity than MF and ER, but higher than CB in the dense mode, and its



(a) (b)

Fig. 11. Comparison of SCFR with different timer schemes. (a) Ferry number with different ps. (b) Message delay with different ns.



(a) (b) (c)

Fig. 12. Comparison of SCFR with different k's. (a) Ferry number with different p. (b) Number of relays with different n. (c) Message delay with different n.

number of relays has a logarithmic upper bound. Based on (2) and (3), SCFR achieves a balanced capacity/delay trade-off among the three metrics.

4. When the initial node distribution is uneven, after a certain number of rounds, a more balanced distribution can be achieved in SCFR, the system enters a stable status, and other performances are improved.
5. In SCFR, a uniform timer helps to reduce the message delay, but causes more ferries in the system.
6. In SCFR, more levels achieve more balanced node distribution and receive smaller wireless transmission range, but causes more ferries, larger message delay, and a larger number of relays.

7 Conclusion

In this paper, we have proposed a logarithmic SCFR protocol in MANETs, which exploits node mobility to assist message delivery especially in likely unconnected networks. SCFR uses a hierarchical structure of mobile nodes called ferries, which effectively accelerates message delivery while supporting ferry sharing. SCFR is adaptive to the traffic rate in the network. From theoretical analysis and performance simulation, SCFR has been proven to have good performance in message moving distance and delay. Meanwhile, SCFR balances the distribution of nodes, which in turn improves the performance of the system.

In future work, we will consider combining the proposed approach to the existing mobility-assisted node redeployment methods. This is motivated by the fact that in the sparse network, the region with no nodes can only be filled if there is traffic going into it. In addition, a certain degree of balance of in and out-traffic is needed. Another future direction is to combine the traditional CB with mobility-assisted routing in MANETs. We will also consider the network model, where traffic originates from each grid instead of each node, as in some sensor network applications.

Acknowledgments

This work was supported in part by US National Science Foundation (NSF) Grants ANI 0073736, CCR 0329741, CNS 0422762, CNS 0434533, EIA 0130806, CNS 0531410, and CNS 0626240.

References

[1] J. Wu and F. Dai, "Mobility-Sensitive Topology Control in Mobile Ad Hoc Networks," *Proc. IEEE 18th Int'l Conf. Parallel and Distributed Processing (IPDPS '04),* 2004.

[2] J. Wu and F. Dai, "Mobility Management and Its Applications in Efficient Broadcasting in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2004.

[3] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-Hoc Wireless Networks," *Proc. IEEE INFOCOM,* 2001.

[4] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory,* vol. 46, no. 2, pp. 388-404, 2000.

[5] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. ACM MobiHoc,* 2004.

[6] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," technical report, Duke Univ., 2002.

[7] W. Zhao, M. Ammar, and E. Zegura, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network," *Proc. IEEE INFOCOM,* 2005.

[8] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm.,* vol. 7, no. 5, pp. 28-34, 2000.

[9] V.W.-S. Wong and V.C.M. Leung, "Location Management for Next-Generation Personal Communication Networks," *IEEE Networks,* vol. 14, no. 5, pp. 18-24, 2000.

[10] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Proc. ACM SIGCOMM,* 1996.

[11] C. Perkins and E. Royer, "Ad Hoc On-Demand Distance Vector Routing," *Proc. Second IEEE Workshop Mobile Computing Systems and Applications,* 1999.

[12] "Delay-Tolerant Networking Research Group," http://dtnrg.org, 2007.

[13] "BAA04-13: Disruption Tolerant Networking (DTN)," technical POC, Preston Marshall, DARPA/ATO, 2004.

[14] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *Proc. ACM SIGCOMM,* 2004.

[15] S. Merugu, M. Ammar, and E. Zegura, "Routing in Space and Time in Networks with Predictable Mobility," Technical Report GIT-CC-04-07, College of Computing, Georgia Inst. of Technology, 2004.

[16] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Network Architecture," Internet Draft: draft-irtfDtnrg-arch-04.txt, DTN Research Group, 2006.

[17] I.F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Ad Hoc Networks J.,* vol. 3, no. 3, pp. 257-281, 2005.

[18] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2006.

[19] J. Davis, A. Fagg, and B. Levine, "Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks," *Proc. Int'l Symp. Wearable Computing,* 2001.

[20] T. Small and Z.J. Haas, "The Shared Wireless Model—A New Ad Hoc Networking Paradigm (Or Where There Is a Whale, There Is a Way)," *Proc. ACM MobiHoc,* 2003.

[21] 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," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '02),* 2002.

[22] "TIER Project," http://tier.cs.berkeley.edu, 2007.

[23] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *Computer,* vol. 37, no. 1, pp. 78-83, 2004.

[24] Q. Li and D. Rus, "Sending Messages to Mobile Users in Disconnected Ad-Hoc Wireless Networks," *Proc. ACM MobiCom,* 2000.

[25] D. Goldenberg, J. Lin, A.S. Morse, B. Rosen, and Y.R. Yang, "Towards Mobility as a Network Control Primitive," *Proc. ACM MobiHoc,* 2004.

[26] B. Burns, O. Brock, and B.N. Levine, "MV Routing and Capacity Building in Disruption Tolerant Networks," *Proc. IEEE INFOCOM,* 2005.

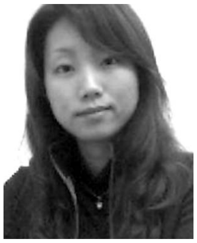
[27] X. Chen and A.L. Murphy, "Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks," *Proc. Workshop Principles of Mobile Computing,* 2001.

[28] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Single-Copy Routing in Intermittently Connected Mobile Networks," *Proc. IEEE Comm. Soc. Conf. Sensor and Ad Hoc Comm. and Networks (SECON '04),* 2004.

[29] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-Hoc Wireless Networks," *Proc. IEEE INFOCOM,* 2001.

[30] S. Cang and J. Wu, "Minimizing Total Communication Distance of a Broadcast on Mesh and Torus Networks," *Proc. First Merged Int'l Parallel Processing Symp. and Symp. Parallel and Distributed Processing (IPPS/SPDP '98),* 1998.

Jie Wu is a distinguished professor in the Department of Computer Science and Engineering at Florida Atlantic University and a program director at the US National Science Foundation. He has published more than 300 papers in various journals and conference proceedings. His research interests include mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He served as program vice chair for the 2000 International Conference on Parallel Processing (ICPP), program vice chair for the 2001 IEEE International Conference on Distributed Computing Systems (ICDCS), and general chair for the 2008 IEEE International Parallel and Distributed Processing Symposium (IPDPS). He is a program cochair for the IEEE First International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '04). He was a coguest editor of a special issue in *Computer* on ad hoc networks. He was also an editor of several special issues of the *Journal of Parallel and Distributing Computing* (JPDC) and *IEEE Transactions on Parallel and Distributed Systems* (TPDS). He is the author of the text *Distributed System Design* (CRC, 1998) and is the editor of the text *Handbook on Theoretical and Algorithmic Aspects of Sensor*, *Ad Hoc Wireless*, and *Peer-to-Peer Networks* (Auerbach, 2005). He served as an associate editor of the *IEEE Transactions on Parallel and Distributed Systems* and is currently on the editorial board of several international journals. He is a recipient of the 1996-1997, 2001-2002, and 2006-2007 Researcher of the Year Awards at Florida Atlantic University. He served as an IEEE Computer Society distinguished visitor and is the chairman of IEEE Technical Committee on Distributed Processing (TCDP). He is a senior member of the IEEE.



Shuhui Yang received the BS and MS degrees in computer science from Jiangsu University, Zhenjiang, China, and Nanjing University, Nanjing, China, in 2000 and 2003, respectively. She is a PhD candidate in the Department of Computer Science and Engineering, Florida Atlantic University. Her current research focuses on the design of localized routing algorithms in wireless ad hoc and sensor networks. She is a student member of the IEEE.

Fei Dai received the PhD degree from Florida Atlantic University and the MS degree from Nanjing University, China. He is an assistant professor in the Department of Electrical and Computer Engineering at North Dakota State University. He has been a senior programmer at Greatwall Computer and a software architect and team leader at J&A Securities, both in China. His research interests include networking, mobile computing, parallel and distributed computing, artificial intelligence, and software engineering.