Available online at www.sciencedirect.com
SCIENCE DIRECT
J. Parallel Distrib. Comput. 63 (2003) 75–86
Journal of Parallel and Distributed Computing
http://www.elsevier.com/locate/jpdc

Communication in disconnected ad hoc networks using message relay

Qun Li* and Daniela Rus

Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA

Received 17 October 2002

Abstract

An ad hoc network is formed by a group of mobile hosts upon a wireless network interface. Previous research in communication in ad hoc networks has concentrated on routing algorithms which are designed for fully connected networks. The traditional approach to communication in a disconnected ad hoc network is to let the mobile computer wait for network reconnection passively. This method may lead to unacceptable transmission delays. We propose an approach that guarantees message transmission in minimal time. In this approach, mobile hosts actively modify their trajectories to transmit messages. We develop algorithms that minimize the trajectory modifications under two different assumptions: (a) the movements of all the nodes in the system are known and (b) the movements of the hosts in the system are not known.

1. Introduction

Mobile computers often disconnect from the network, and when they reconnect, they might find themselves with a radically different network connection in terms of bandwidth, reliability or latency. Approaches to cope with the transmission of data in mobile, wireless networks include traditional techniques such as try, timeout, sleep, retry, etc., and wireless routing algorithms. The simple try, timeout, sleep, retry loop can fail particularly if the system does not happen to retry connection during a brief reconnection period. The current wireless networking solutions are not sufficient, because an entire path to the destination machine has to be available. Suppose you want to transmit data from machine M_s to machine M_g and the path includes at least one intermediate node, say machine M_i (this is often the case in wireless networks because of range limitations.) In order for the transmission to be successful, the connections between M_s and M_i and between M_i and M_g must be available at the same time. The probability of this event is much smaller than the probability that one of the two hops (from M_s to M_i or from M_i to M_g) is open.

We propose algorithms for active communication in ad hoc wireless networks. Previous research in this area has concentrated on fully connected networks, in which any two hosts can communicate with each other directly or via other intermediate hosts. In an ad hoc network, the hop by hop communication may not be possible because the neighboring hosts may be disconnected. Instead of statically waiting for network reconnection, a host can actively change its location to achieve connectivity using knowledge about the location of other hosts. We believe that such active message transmission is feasible when the hosts in the network cooperate for a joint mission, and useful for applications that require urgent message delivery.

In this paper we explore the possibility of changing the host trajectories in order to facilitate communication. We show how information about the motion of the destination host can be used to determine how messages can be sent to this host by cooperating intermediate hosts. Given an ad hoc network of mobile computers where the trajectory of each node is approximately known, we would like to develop an algorithm for computing a trajectory for sending a message from host A to host B by recruiting intermediate hosts to help. In our context, recruiting means asking intermediate hosts to change their trajectory in order to complete a routing path between hosts A and B. We would like to minimize the trajectory modifications while getting the message across as fast as possible.

Two algorithms are studied in this paper. In the first algorithm, we assume the information about the motions and locations of hosts is known to all hosts,

*Corresponding author.
E-mail addresses: liqun@cs.dartmouth.edu (Q. Li), rus@cs.dartmouth.edu (D. Rus).

0743-7315/03/$ - see front matter © 2002 Elsevier Science (USA). All rights reserved.
doi:10.1016/S0743-7315(02)00033-3

or can be estimated within some error parameters. The second algorithm does not assume that the movement of the hosts is known.

This approach to message transmission can be implemented using mobile agents [1]. A mobile agent is a program that can migrate under its own control. The main advantage of using mobile agents for communication in ad hoc networks is that they can function as "wrappers" on messages. The mobile agent wrapper (called an active message) provides a certain level of autonomy for messages and allows them to reside at intermediate points in the network. This enables a message to propagate itself to the destination incrementally, which is an advantage over traditional message transmission approaches in which the entire path from the starting location to the destination must be available. Thus, the communication protocol we propose is an application-layer protocol (rather than a network-layer protocol.) While the network cannot route a message to the destination due to a network partition, it will try to do an "up-call" for the scheme we present in this paper. A program can determine the moving route of the hosts relaying the message. Other application programs, for example a controller can then decide if the route for the message makes sense or if there are better approaches. For example, in a tactical robotic network where a team of robots is deployed to perform sensing tasks, the message routing program could suggest trajectory modifications for the team, while the individual robots could decide the ultimate host trajectories.

This paper introduces the idea of message relays but does not address all the technical issues. Many questions have to be answered in order to completely characterize the applications for which this approach to communication is suitable. Our goal here is to show that active message transmission by relay is a promising protocol for communication in ad hoc wireless networks. We hope that this work will stimulate more research towards understanding this concept.

The remainder of this paper is organized as follows. Section 2 introduces the related work. The message transmission algorithm with full knowledge of the host motions is described in Section 3. Section 4 presents the performance evaluation of the algorithms when they operate with imprecise information about the hosts' locations. The message transmission algorithm without full knowledge of the host motions is analyzed in Section 5. Section 6 discusses experiments.

2. Related work

We are inspired by recent progress in three areas: ad hoc networks, global position system (GPS) location information aided routing, randomized routing, personal communication systems (PCS), and mobile agents.

There has been a lot of research on routing in ad hoc networks [2–8]. Routing algorithms have to cope with the typical limitations of wireless networks: high power consumption, low wireless bandwidth, and high error rates. All these routing protocols assume that the network is connected. The work described in this paper is different in that our networks may be disconnected.

Algorithms that make use of the GPS location information to aid route discovery can be found in [7,9–11]. Ko [9] proposes a location-aided routing protocol in which a node forwards a message to its neighbors by using GPS location information. Bose [10], Stojmenovic [12], and Karp [7] propose some routing protocols that can guarantee message delivery by using location information. Boukerche [13] propose GZRP, a protocol that combines the zone routing protocol (ZRP) scheme and GPS.

Boukerche et al. [14,15] propose a randomized congestion control scheme for the DSDV routing protocol. Each node has some probability of propagating the routing information. When the routing information originating from a node is diffused slowly, the load on that path decreases. They also present a very nice analytical model based on Markov chains. A different randomized protocol [16] uses mobile agents to help routing. Other interesting routing protocols include power-aware routing [17,18].

Another related area is PCS location management [19–21]. Most location management techniques use a combination of updating and finding, in an effort to select the best trade-off between the update overhead and the delay incurred searching. Specifically, updates are not usually sent every time a host enters a new cell, but rather are sent according to a pre-defined strategy, for example restricting the searching operation to a specific area.

3. Message transmission in known mobile networks

In this section we develop an algorithm for message transmission in a dynamic ad hoc network that uses a strong assumption: the moving trajectories of all the nodes in the system are known. The assumption holds for many applications, especially when the hosts move along existing roads and highways: a police car follows the road at a constant speed, a soldier patrols on the beat, and rescuing crews move according to detailed plans. In subsequent sections we show less restrictive generalizations of this scheme.

We propose a communication scheme in which a message reaches its destination even when the destination host is out of range. Rather than waiting for a connection from the originator to the destination (which may never become available), we propose a scheme in which hosts *actively move* to relay messages. We would like to minimize the movement necessary to relay a message.

Algorithm 1. The behavior of each host h_i in an ad hoc network where message relays are used for communication.

1: **for** each host h_i in the system pursue investigation while waiting to receive messages. generate message when needing to communicate **do**
2:     **if** a message m_j is received **then**
3:         **if** the recipient of m_j is h_i **then**
4:             process m_j
5:         **else**
6:             **if** the recipient of m_j is h_k **then**
7:                 compute *Optimal_Relay_Path* (h_i, h_k), given as a list of tuples of (host, path-to-reach-host); send the message to the head of this list (this may involve a trajectory modification to get within transmission range from this head node, followed by return to the original trajectory)
8:             **end if**
9:         **end if**
10:     **end if**
11: **end for**

3.1. The case of multiple messages

Suppose a set of hosts move according to pre-specified trajectories and the maximal speed of the hosts is high as compared to the distance between hosts. Hosts proceed with their mission and occasionally deviate to relay messages. We are especially interested in applications where the network is almost connected; the distance between two adjacent hosts is slightly larger than the transmission range. In such situations, the time for a host to get into communication range is quite short, and it does not affect its location estimation by the other hosts very much. The time spent by a host deviating from the original trajectory is not too large, although it does give rise to error on location estimation (In some applications, for example on a battlefield, back-up devices such as walkie-talkies can be used to update the location information and thus correct the error introduced by trajectory changes.).

We assume that each host in the system has a task to carry out. That task may include information processing and moving. Occasionally, hosts need to send each other information. Thus, we can model the behavior of this system as a basic loop (Algorithm 1). The interesting component of the loop is the else-if block of the algorithm (line 6). When the host needs to transmit a message to someone out of range, it computes a sequence of intermediate hosts that can relay the message to the destination, where each intermediate host modifies its trajectory in the smallest possible way. The sequence of hosts and path modifications can be computed since all the host movements are known. In the next section we will detail this computation.

For the rest of this section, we focus on applications where the message transmission rate is low, and each trajectory modification is very small. Message flow can be modeled as a single message flowing in the system at a time, so that no trajectory interference, and deadlocking will not occur.

3.2. The case of a single message

In this section we assume that all the hosts' motion descriptions are known. We describe a communication algorithm suitable for the following types of distributed applications: if the maximal possible speed of the hosts in the system is larger than the moving speed of the message recipient, the message can be sent successfully given the moving descriptions of the hosts.

Suppose h_1, h_2, h_3, h_4 are four mobile hosts in an ad hoc network (see Fig. 1) with known motions at dispatch time. If h_1 wants to send a message to h_4 and h_4 is not within transmission range, h_1 needs to get closer to h_4. Host h_1 may move all the way to the transmission range of h_4 to send the message directly, but this movement may be too expensive. If the distance between h_1 and h_4 is too large, h_1 can approach another host h_2 by moving a short distance and relaying the message to h_2. After that, h_2 can do the same until the new host is within the transmission range of h_4. By using intermediate hosts, the message transmission time may be shorter than that of the method which forces h_1 to move all the way to h_4 approach h_4. Thus, our problem is, given a mobile ad hoc network, (which may be disconnected,) and the motion descriptions of the hosts, find the shortest time strategy to send a message from one host to another.

The intuition for computing the Optimal Relay Path is as follows. Using knowledge about the trajectories of h_2, h_3, h_4, host h_1 can compute the trajectories with the shortest time to approach h_2, h_3, h_4 (we describe this algorithm in Section 3.2.1). The shortest trajectory (say

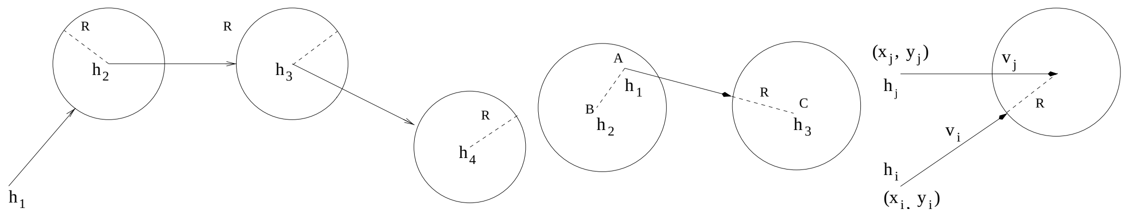

Fig. 1. In the first figure, node h_1 sends a message to h_4 by way of intermediate hosts h_2 and h_3. Disks corresponds to the transmission range of hosts and arrows show approach trajectories to relay messages. The second and the third figure describe two examples of approaching hosts. The disks represent the transmission range. In the second figure, all hosts are static. Host h_1 can send messages to h_2 directly. But host h_1 must move toward h_3 to establish communication. The third figure shows the optimal trajectory of h_i to approach h_j starting from (x_i, y_i), while h_j is moving to the right starting from (x_j, y_j).

to host h_2) may provide a faster way of reaching the transmission range of the other hosts. The shortest trajectories can be computed incrementally using increasingly more intermediate hosts. The Optimal Relay Path can be formalized under the following assumptions: (1) Two hosts can communicate with each other within range R; the size of R depends on the communication hardware. (2) If host h_1 wants to send a message to host h_4, who is out of the range, h_1 can move some distance and send the message to h_4, or it can approach an intermediate host that can act as a relay to send the message to h_4. For example, in Fig. 1 (first), h_1 moves to approach h_2, h_2 moves to approach h_3, then h_3 moves and sends the message to h_4. (3) Only one message at a time circulates in the system.

Before presenting the Optimal Relay Path algorithm, we introduce the following terminology. The motion of a host h_i is *predictable* if there is a known function $P_i(t)$ which describes the position of host h_i at time point t, prior to changing its trajectory. *A moving path* from A to B is a sequence of hosts, $h_0, h_2, \ldots, h_k$ (where $h_0 = A$ and $h_k = B$) with their moving strategy which gives how h_i moves to approach h_{i+1} to send a message. In first figure of Fig. 1, $h_1h_2h_3h_4$ is a moving path from h_1 to h_4. *An optimal path* from host A to host B is a moving path of hosts which gives the least time to send the message from A to B.

Algorithm 2 describes the Optimal Relay Path algorithm, which determines the shortest path to the destination of the message. The algorithm computes the direct path from h_0 to other hosts in the initialization part. The main body consists of choosing the host reachable in the minimal time among the hosts which have not been processed, and marking the host ready. Then the current minimal time from h_0 to all hosts that are not ready are updated. The running time of the algorithm is $O(n^2t)$ where t is the running time of the algorithm *Optimaltrajectory*.

3.2.1. Finding the optimal trajectory for relaying a message

Suppose $P_j(t)$ is the position of host h_j at time point t, and the initial time point when host h_m begins to approach h_j is t_0. The following two equations give the optimal strategy for host h_m to approach h_j (Recall that the moving speed is known.) More precisely, by solving the equations, the velocity of host h_m and the approaching time can be obtained. In these equations, $P_i(t)$ denotes the trajectory of host h_i, v is the moving speed of the host, and t denotes time

$$|\vec{P}_j(t) - (\vec{P}_m(t_0) + \vec{v} \cdot (t - t_0))| \leqslant R, \tag{1}$$

$$\frac{\vec{P}_j(t) - \vec{P}_m(t_0)}{|\vec{P}_j(t) - \vec{P}_m(t_0)|} = \frac{\vec{v}}{|\vec{v}|}. \tag{2}$$

Algorithm 2. Sketch: the Optimal Relay Path to all hosts in the system. Input: (1) initial time when host h_0 begins to send a message, (2) the moving function of host h_i, which gives the position of h_i at time t. Output: the optimal moving path from host h_0 to all other hosts $h_1, h_2, \ldots, h_n$.

1: Compute the optimal trajectory for host h_0 to reach all the other hosts directly, record the earliest time point $t[k]$ for h_k.
2: Choose the unmarked host h_i with the least $t[i]$, mark h_i, $Ready[h_i] = 1$.
3: Compute the optimal trajectory (use the OptimalTrajectory algorithm) for host h_0 to reach all the unmarked hosts, such as, h_j by way of h_i. If the time point computed for the optimal path from h_0 to h_j by way of h_i is less than the original $t[j]$, update $t[j]$ with the newly computed time point
4: Goto 2 until all the hosts have been marked

Eq. (1) gives the condition for host h_j to be in the transmission range of host h_m when h_m approaches h_j. Eq. (2) gives the condition for host h_m to move to the direction of host h_j at time point t. The equations have been derived using elementary geometry.

Eqs. (1) and (2) lead to the algorithm *Optimaltrajectory* for computing optimal trajectories. The solution depends on the movement of the hosts in the system. (The middle and the right images in Fig. 1 describe two examples.)

3.3. Analysis

We now show that if the motions of all mobile hosts are predictable, and there is only one message circulating in the system, then the Optimal Relay Path algorithm computes the optimal communication paths from one given host to all other hosts in the system.

Lemma 3.1. *In an environment with only two hosts, if the motion of host h_j is predictable, there is an optimal moving path from h_i to h_j given the description of h_i (the initial position and the maximal speed).*

Proof. Recall that *Optimaltrajectory* computes the optimal moving path from h_i to h_j at time point t_0, for a given moving description. □

Lemma 3.2. *Suppose $h_0, h_1, \ldots, h_{b-1}, h_b$ is an optimal path from h_0 to h_b. After h_{b-1} receives the message relay originating at h_0, it moves according to the path given by algorithm Optimaltrajectory.*

Proof. If Lemma 3.2 does not hold, we can replace the path of h_{b-1} with the path given by the algorithm *Optimaltrajectory*. We get a better moving path, which is a contradiction. □

Lemma 3.3. *If $h_0, h_1, \ldots, h_b$ is an optimal moving path from h_0 to h_b, then $h_0, h_1, \ldots, h_{b-1}$ must be an optimal path from h_0 to h_{b-1}.*

Proof. Suppose Lemma 3.3 does not hold. Then $h_0, h_1, \ldots, h_{b-1}$ is not the optimal path from h_0 to h_{b-1}. We can substitute the path from h_0 to h_{b-1} with the optimal path from h_0 to h_{b-1}. It is easy to see we get a better path from h_0 to h_b, which is a contradiction. □

Theorem 3.1. *The Optimal Relay Path algorithm (Algorithm 2) gives the optimal moving paths from host h_0 to all other hosts.*

Proof. Proof by mathematical induction.

Let $h_0, h_1, \ldots, h_{n-1}$ be the host sequence in the order in which their *Read* y variable has been marked.

1. Initially, *Read* $y[h_0]$ is marked. The optimal moving path from h_0 to h_0 is the node itself.
2. Suppose after *Read* $y[h_i]$ is marked, the algorithm gives the optimal moving paths from h_0 to h_1, $h_2, \ldots, h_{i-1}$, and h_i.
3. Consider the time when *Read* $y[h_{i+1}]$ becomes marked. We get a moving path with the minimal time among all the moving paths (from h_0 to h_{i+1}) which consists of only the marked hosts.

We first show that the path computed is the optimal moving path if we only consider the marked hosts. By Lemma 3.3, the optimal moving path from h_0 to h_b can be divided into two parts: (1) the optimal moving path from h_0 to h_{b-1}, and (2) the optimal path in which h_{b-1} moves to approach h_b and sends the message to h_b. By induction, we have all the optimal moving paths from h_0 to $h_1, h_2, \ldots$, and h_{b-1}. Thus, by analyzing the algorithm (the essence of the algorithm is to enumerate all possible paths to reach h_b via $h_0, h_1, \ldots, h_{b-1}$), we get the path with the minimal time, which must be the optimal moving path among all the marked hosts.

We now show that the path found is the optimal moving path from h_0 to h_{i+1} if we consider the entire system. If the path is not optimal, the optimal moving path must consist of some unmarked hosts. Let h_k be the first such host in the path. In the optimal moving path, the time for the segment from h_0 to h_k, which only consists of the hosts which have been *Read* y, is less than the time of the moving path from h_0 to h_{i+1}. So h_{i+1} does not have the minimal time, which is a contradiction. □

4. Message transmission under location error

An important property of the Optimal Relay Path algorithm (see Algorithm 2) is that it works even if the location of the hosts in not known precisely—that is, the trajectories are specified within certain error parameters. This is an especially useful property for real applications (for example involving moving cars and robots) where uncertainty in the location information is a fundamental component (movement modifications are likely to contribute to errors in the host location estimations.) In this section we examine the performance of the Optimal Relay Path algorithm for routing and relaying messages in the presence of error. We assume that the location estimates are specified within known error bounds r. We derive an upper bound for trajectory changes for message relays. In other words, we compute the sum of the distances traveled by each host involved in the transmission of one message. The exact computation of the traveled distance is not sufficient because the location of hosts is known only approximately, and extra time might have to be spent identifying exactly where the host is.

Suppose the movement of each host is restricted to a region of radius r we call *scope*. Such a restriction is realistic when the moving speeds of the hosts are relatively slow. If we estimate that a host is static at the center of the scope, the error of the estimation is at most r. The upper bound for the total movement necessary to relay a message is given by the following result:

Theorem 4.1. *In an environment with location error, suppose the estimated moving description of host h_i is static at O_i. Then the sum of the length of the moving path computed by Optimal Relay Path is at most $(4n-5)r$ more than that of the optimal moving path, where n is the number of the hosts in the system, and r is the maximal error.*

In order to compare the path we get with location error to the optimal path, we take as reference an imaginary system in which all hosts are static. We compare the path in this imaginary system with the path obtained in our algorithm and the optimal path separately. Thus, we get the difference between the length of the path in our algorithm and that of the optimal path. The proof for Theorem 4.1 is based on Lemmas 4.1 and 4.2.

In order to decrease the error of the location estimation procedure, the previous algorithm can be refined by adding the exchange of up-to-date location information of hosts when two hosts are within the transmission range of each other for message transmission. The motion estimation of a host can be organized as many tuples, each of which corresponds to one host in the network. The tuple structure is: $\langle host_{id}, time, location, velocity, motion\_description \rangle$. The first four items denote the host's location and velocity at time point t. The motion description, which is system dependent, is used to describe the characteristics of the host's motion, e.g., the host is moving between A and B, back and forth. When the hosts are close enough to exchange messages the motion information about the hosts is also exchanged. The motion information of a host will be updated according to the latest information. Thus, the up-to-date motion information of the hosts will be propagated.

When the maximal possible speed of hosts is high and the distance between the two hosts is a little longer than the transmission range, the time spent on message transmission is not substantial. In a network that is almost connected, the error is small if the approach speed is high.

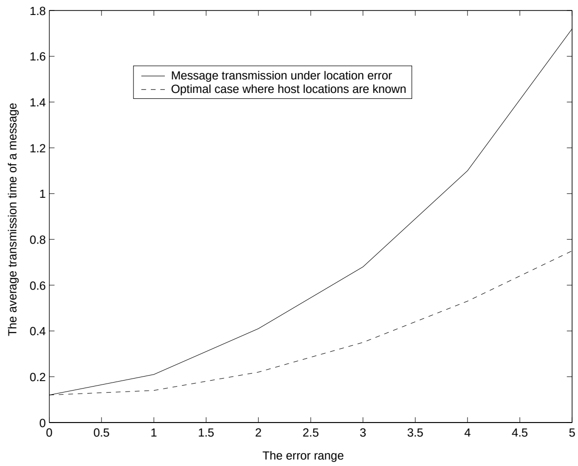

Fig. 2. This graph shows the comparison in simulation between message transmission under location errors and the optimal case in which all host locations are known. The x-axis denotes the error range—that is, the maximal error of the guessed the locations. The y-axis denotes the average time to transmit a message from its origin to its destination. The simulation was done with 20 hosts, a network space of 20×20, maximal moving speed for each host of 5, transmission range of 5, and a message arrival rate of 0.2, 0.1, and 0.02 for each host (we average the transmission time). The simulation was run for 10^4 time units. For the x-axis, 0 corresponds to the case when the hosts are static, and 1 corresponds to the case when hosts can be anywhere in a circle of radius 1 centered on their guessed location.

Fig. 2 gives simulation data for how the location error affects the performance of the system. The larger the location error is, the more time is spent approaching another host.

Next, we want to analyze the performance of the algorithm if the maximal location error is $r \leqslant R$, that is, if any host h_i in the system never moves beyond the range of a circle with center O_i and radius r.

Let us consider three scenarios (see Fig. 3). In the first scenario, all the hosts are stationary at (x_i, y_i) (for all $1 \leqslant i \leqslant n$) and Algorithm 2 knows the precise location of each node. In the second and third scenarios, each host, say h_i can be anywhere in a circle centered at (x_i, y_i) with radius r. In the second scenario, the location of host i is estimated at (x_i, y_i), while the third scenario assumes all the locations are known precisely by all the hosts.

Lemma 4.1. *Suppose the length of the moving path in scenario* 1 *and* 2 *are* l_1 *and* l_2, *respectively. We have* $l_2 - l_1 \leqslant (2k-3)r \leqslant (2n-3)r$.

Proof. As Fig. 3 (left) illustrates, h_j starts to approach h_i from A. When h_j reaches B, h_i may be in any position in the range of the circle centered at C with a radius of r.

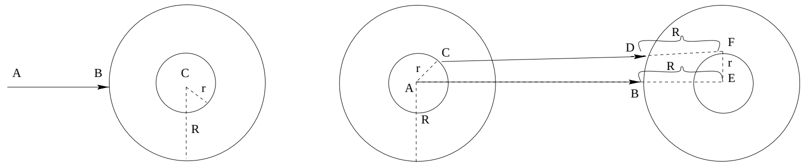

Fig. 3. Two hosts in an environment with location error. The bigger disks represent transmission range (R), and the smaller ones are moving scope (with maximal error r). In the first figure, h_j starts from A to approach h_i. The second figure compares CD, the optimal path of h_j and AB, the path computed by our algorithm 2 using the estimated locations.

So h_j can move at most r before it can send the message to h_i (an extra move). After h_i receives the message from h_j, it begins to approach the next host. Host h_i can first go back to C (another extra move) before it approaches the next host. In this period, the distance traveled by host h_i is r more than the distance traveled when the location information is exact. The same holds for h_j. For the path $h_1 \rightsquigarrow h_k$, we have $2(k-1)-1$ extra moves, thus we have $l_2 - l_1 \leqslant (2k-3)r \leqslant (2n-3)r$. □

Lemma 4.2. *Suppose the length of the moving path in scenario* 3 *is* l_3. *We have* $l_1 - l_3 \leqslant (2n-2)r$.

Proof. In Fig. 3 (right), the large circles give the transmission range, while the small circles give the movement range. Let the real position and the estimated position of h_j be C and A, and the position and the estimated position of h_i be F and E. Now suppose h_j needs to approach h_i in an optimal path in scenario 3. Then CD is a fragment of the moving path. In scenario 1, the positions of h_j and h_i are A and E. The moving path for h_j to approach h_i is AB.

Let $A = (0,0)$ and $E = (L,0)$. B is a point on line AB such that $|BE| = R$. C and F are two points within the circles of radius r centered at A and E, respectively. D is a point on line CF such that $|DF| = R$. Then the maximal value of $|CF|$ is achieved when C and F are on the circles centered at A and E, respectively. By applying basic trigonometry, we get $|CD| + R \leqslant |AB| + R + 2r$.

If $P_3 : h_1 \rightsquigarrow h_k$ is a moving path in scenario 3 whose sum of distances traveled is l_3, then in scenario 1 we have a moving path $P_4 : h_1 \rightsquigarrow h_k$ that has the length l_4 such that $l_4 - l_3 \leqslant (2k-2)r \leqslant (2n-2)r$. Because l_1 is the length of the shortest path among all moving paths from h_1 to h_k in scenario 1, we have $l_1 \leqslant l_4$. Thus we have $l_1 - l_3 \leqslant (2n-2)r$. □

By Lemmas 4.1 and 4.2, we have $l_2 - l_1 \leqslant (2k-3)r \leqslant (2n-3)r$ and $l_1 - l_3 \leqslant (2n-2)r$. In summary, $l_2 - l_3 \leqslant (4n-5)r$, and thus Theorem 4.1 holds. This analysis describes the trade-off between the amount of error we would like to tolerate in our relay algorithm and the extra movement hosts have to make due to location estimation errors.

5. Message transmission in unknown mobile networks

When the error of the estimated location is smaller than the transmission range, the previous algorithms work well. But the error can be large if random factors distract the motion of a host from the estimated track. When the error is larger than the transmission range, tracing hosts according to the previous schemes is impossible. In this section we present a method that makes it possible to communicate to all hosts in the system despite their unknown movement.

We propose a method in which hosts inform the other hosts of their current position. The key issues that need to be considered to make this approach work are: (1) when a host should send out information about its location update; (2) to whom the host should send out this information; and (3) how the host should send out this information. In this section we present solutions to (1) and (2). (3) can be implemented using a walkie-talkie, satellite, or wireless modem hardware.

We assume that each host is confined to movement within a region we call *scope* and each host knows who is the host that keep track of its location we call *tracking host*. Location updates must occur when the host leaves its current scope. If the radius of the scope is less than the transmission range, then we can guarantee that the host can be found by its tracking host since the tracking host can go to the center of the scope and broadcast a message.

We model the communication problem in unknown mobile network environments by constructing a minimum spanning tree. Let G be a weighted graph whose vertices correspond to the hosts in the system. The edges of G connect hosts to tracking hosts. The edge weights correspond to the physical distances between the hosts. The minimum spanning tree of G contains the shortest edges in the graph that provide full connectivity in the graph.

The neighbors in the minimum spanning tree provide the communication routes for messages. Each host has the responsibility of updating its location by informing all the hosts connected to it in the minimal spanning tree. Thus, when a host leaves its scope, it needs to inform only its neighbors in the minimum spanning tree. It is clear that there is a trade-off between the size of the host's scope and the frequency of its location update messages. We would like to quantify this trade-off in the next section. Algorithm 3 gives the algorithm for the location update in this communication method.

updates. A shorter scope radius leads to more frequent updates, because the host is more likely to move out of scope. We would like to compute this trade-off to identify the most optimal scope size.

Algorithm 3. This algorithm shows pseudo-code for the location updates when hosts do not know a priori their moving paths. **Notations:** $\mathbf{t_i}$: the latest time when h_i got the location update of h_0. $\mathbf{t}$: the current time. $(\mathbf{x_{t_i}}, \mathbf{y_{t_i}})$: the location of h_0 at time t_i. $(\mathbf{x_t}, \mathbf{y_t})$: the location of h_0 at time t.

```
for all hosts h_1, h_2, ..., h_k that are adjacent to h_0 in the minimum spanning tree
do
    Compute the optimal radius r_i between h_0 and h_i.
    if |(x_t, y_t) - (x_{t_i} + v_x(t - t_i), y_{t_i} + v_y(t - t_i))| >= r_i then
        move to h_i to update its location (t_i = t and (x_{t_i}, y_{t_i}) = (x_t, y_t)).
    end if
    if there is message exchange between h_0 and h_i then
        update the t_i and (x_{t_i}, y_{t_i}) to the current time and location
    end if
end for
```

5.1. Communication in the minimum spanning tree

In this section we analyze the trade-offs between scope and update frequency in the minimal spanning tree protocol, by considering the error in a host's estimation about the location of another host. We first consider a two-node system. Our result for the two-node system can be used to compute the optimal location error for a multi-node system connected by the topology of its minimum spanning tree.

For simplicity, we assume that hosts maintain their neighbors throughout the experiment (that is, the topology of the minimum spanning tree does not change.) Extensions to dynamically changing minimum spanning trees can be done using the previous algorithms for dynamically constructing a minimum spanning trees derived in [22,23].

Suppose there are two hosts which have to communicate with each other, but they are out of transmission range. There are two types of message exchanges: (1) an actual message and (2) a location update message. Each host has its own task to carry out which may require movement. We would like to identify the optimal scope size with respect to how much the hosts need to travel in order to communicate with each other. Suppose host h_i needs to communicate with h_j and h_i and h_j are neighbors in the MST. Thus, they need to keep track of each other's locations. If the scope size is small, h_i has a good idea of where h_j actually is, but h_j will have to update its location more frequently. If the scope size is large, h_j has to do fewer location updates, but h_i has a less good approximation for where h_j is so h_i has to travel more in order to communicate. There is a trade-off between the length traveled by a host to communicate with another host and the frequency of location updates. A shorter scope radius leads to more frequent updates, because the host is more likely to move out of scope. We would like to compute this trade-off to identify the most optimal scope size.

Since the motion variance of each host, that is, the uncertainty of a host's location increases in time, a good model for the time-varying behavior of a mobile host is the Brownian motion with a drift process [20]. The two-dimensional Brownian motion with a drift process can be described by the distribution

$$p_{xy}(x, y|x_0, y_0, t) = \frac{1}{2\pi\sqrt{D_x D_y}(t - t_0)} \times \exp\left(\frac{-[(x - x_0) - v_x(t - t_0)]^2}{2D_x(t - t_0)} + \frac{-[(y - y_0) - v_y(t - t_0)]^2}{2D_y(t - t_0)}\right), \quad (3)$$

where (x_0, y_0) is the initial location of the host, (v_x, v_y) are the components of the drift velocity along the x and y axes, t_0 is the initial time, and (D_x, D_y) are the diffusion parameters with unit of $(length^2/time)$. Large (v_x, v_y) correspond to rapid location changes. The uncertainty of the location is determined by (D_x, D_y). Large uncertainty corresponds to larger scope for the location of the host.

Without loss of generality, suppose $D_x = D_y = D$. From Eq. (3), a radius r of a scope within which the probability of a host is equal to γ at time t can be expressed as: $r(t) = \sqrt{2D(t - t_0)\ln(1/(1-\gamma))}$. The center of the scope is at $(v_x(t - t_0), v_y(t - t_0))$.

Suppose we have two hosts h_1 and h_2. Currently, the distance between h_1 and h_2 is l ($l \geqslant R$), and the rate of messages transmitted between h_1 and h_2 is λ. We want to find the optimal radius of the motion scope. We assume that the maximal possible speed of a host is quite large compared with the host's general moving speed. Thus, the host does not need to consider the effect of the message transmission or the location updating time.

Let r be the radius of the motion scope ($r \leqslant R$). The host will stay in the scope with radius r with probability γ until time t_r. Thus, the average distance for the host travels to transmit messages and updates locations in a unit time is

$$Y = \left(\lambda + \frac{1}{t_r}\right)(l - (R - 2r)), \tag{4}$$

where $l - (R - 2r)$ is the maximal distance for the host travels to approach another host according to the analysis in Lemma 4.1. We want to minimize the location update Y subject to $r \leqslant R$. The following result shows that Y can only obtain its minimal value at three distinct locations.

Theorem 5.1. *The minimal value of the average distance traveled by two hosts to transmit messages and location updates occurs at one of three possible values for* r: $2(P(l-R)/2\lambda)^{1/3}$, $2d^{1/3}\cos\theta/3$, *or* R.

Proof. We know $t_r = r^2/2D\ln(1/(1-\gamma))$. Let $P = 2D\ln(1/(1-\gamma))$, we have $Y = 2\lambda r + 2P/r + P(l - R)/r^2 + \lambda(l - R)$. Take the derivative of r on the both sides of the equation and set the derivative to zero. We get a cubic equation. To solve the equation, let $d = \sqrt{P^3/27\lambda^3}$, and $\theta = \cos^{-1}(3/2(l-R)\cdot\sqrt{3\lambda/P})$. Thus, the minimal value of Y can only be obtained when r is $2(P(l-R)/2\lambda)^{1/3}$, $2d^{1/3}\cos\theta/3$ or R. $\square$

Since there are three possible places for attaining the minimum value for r, we would like to experimentally study when exactly the optimum happens. Fig. 4 shows the solution for the optimum radius (defined by Eq. (4)) for different parameters. We denote by k the ratio between the distance of the two hosts and the transmission range, λ the message arrival rate, D the diffusion parameter, $m = D/\lambda$ the ratio between D and λ.

Fig. 4 (left) describes the change of the optimal radius as m grows. The curves are plotted with for $k - 1 = 8, 4, 2, 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$ and $\gamma = 95\%$. Fig. 4 (right) shows the optimal radius change with the change of the $k - 1$. It includes five curves with $m = \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$. Except for the $m = 1/8$ curve, the others are not very smoothly connected. The reason is that the optimal radius may take one of the three values according to the different k. When k is small, the optimum radius is $2\,(P(l-R)/2\lambda)^{1/3}$; as k increases, it becomes $2\,d^{1/3}\cos\theta/3$; when k is quite large the optimum radius becomes R.

The distance traveled by the hosts is determined by the length of a single trip and the number of trips. Fig. 4 (right) shows that the bigger k is, the longer the optimal radius is. The reason is that for a large k (that is, a large distance between two hosts), reducing r will be less important than reducing the number of trips traveled by the hosts in a unit of time. The ratio m affects length in the similar way. When D is small, the time for a host to go beyond the fixed scope is long, so the optimal radius should be small. On the other hand, when λ is small, the location updates will be dominant. Thus, reducing the number of location update trips, that is, increasing the location update period, is better. As a result, the optimal radius should be bigger for a small D.

6. Simulation experiments

We have developed a simulation system to study our algorithms. We focus on evaluating how message relaying interferes with a host's task. We use three metrics for this evaluation: the percentage of the average working time, the ratio between the standard deviation of the working time and the average working time, and

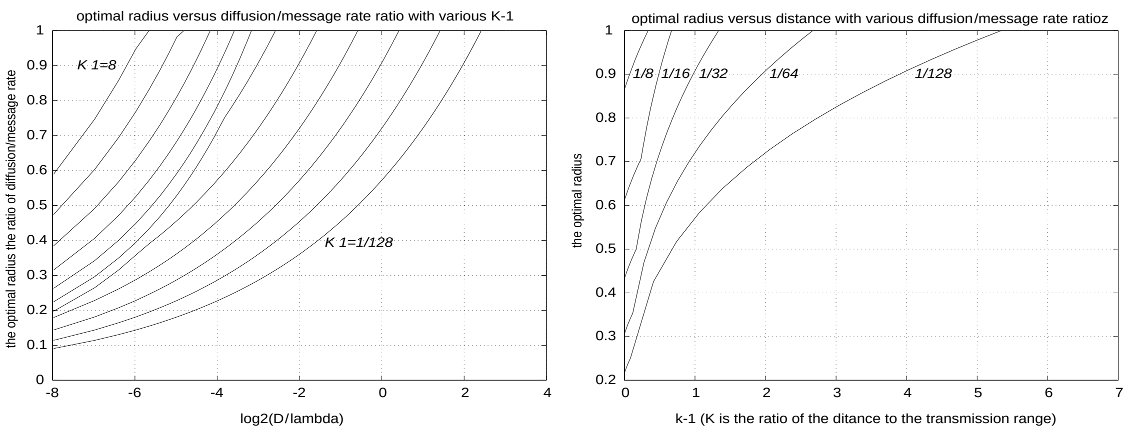

Fig. 4. This figure shows the optimal radius of the scope for the hosts. The left figure shows the dependency of this radius (represented by the y-axis) on the ratio D/λ. Each curve is drawn for different values of k, the ratio defined by distance between two hosts, divided by the transmission range. The right figure shows the dependency of the optimal radius (the y-axis) on k. Each curve is drawn for a different value of D/λ.

the average transmission of a message. The first metric denotes how much time the hosts spend on their own work instead of message transmission. The second metric evaluates how balanced are the host workloads. The third metric measures how fast a message can be sent. We assume that the transmission time can be ignored if two hosts are within transmission range. Thus, the message transmission time is the sum of the host's movement time and any possible waits.

We examine our metrics by varying five parameters: the scope of the network space (that is, the total area where the experiment is done), the number of hosts, the transmission range of each host, the moving speed of each host, and the message arrival rate of each host. We assume all hosts have the same transmission range, moving speed, and message arrival rate. Each host generates messages according to a Poisson distribution. The message recipients are generated randomly and messages are transmitted according to the Optimal Relay Path (Algorithm 2) algorithm, which computes the itinerary for a message. We have done two types of experiments.

Instantaneous message transmission: In this experiment message transmission has the highest priority. Thus, upon receiving a message for relay, the host stops its current task and goes to the next host in the itinerary to transmit the message. Upon return to its original location, the host first checks for waiting messages and only if there are no waiting messages it resumes executing its task.

Delayed message transmission: In this experiment, message relaying is delayed in favor of the host's task for some amount of *waiting time*, which is a parameter to the experiment. We use a waiting time vector whose components correspond to waiting times for all the hosts. We design the waiting time vector according to our network topology in this experiment. This experiment was designed to increase the percentage of the time hosts devote to their tasks. All messages accumulated at a host in the waiting period are sent to the next host as a group if their next destination is the same. Figs. 5–7 show the data we compiled from these experiments.

Fig. 5 demonstrates the effect of varying the waiting time of hosts. Typically, the working time increases with larger waiting times. With a larger waiting time, more messages are accumulated, thus some messages may be sent together. The average message transmission time also increases with a higher waiting time. If the focus is to increase the percentage of working time and ratio between deviation and average working time, delayed message transmission is always better than instantaneous message transmission. We also observed that the percentage of working time stays the same beyond a certain level of waiting, which provides empirical support for choosing a good value for the waiting time for real applications.

Fig. 6 shows the comparison between instantaneous message transmission and delayed message transmission while the transmission range is changed. As the transmission range increases, the working time increases, and the average message transmission time decreases. The larger transmission range contributes to ashorter travel path, which in turn affects the message transmission time and working time. We note that delayed message transmission does much better than instantaneous message transmission with respect to the percentage of working time and the ratio of deviation and average working time.

Fig. 7 shows the influence of the various maximal speed values of the hosts on performance. It is obvious that a larger speed improves performance.

In addition to the quantitative results, we observed the following qualitative behavior in our experiments:

- The percentage of the time spent on message transmission is larger if the message arrival rate is

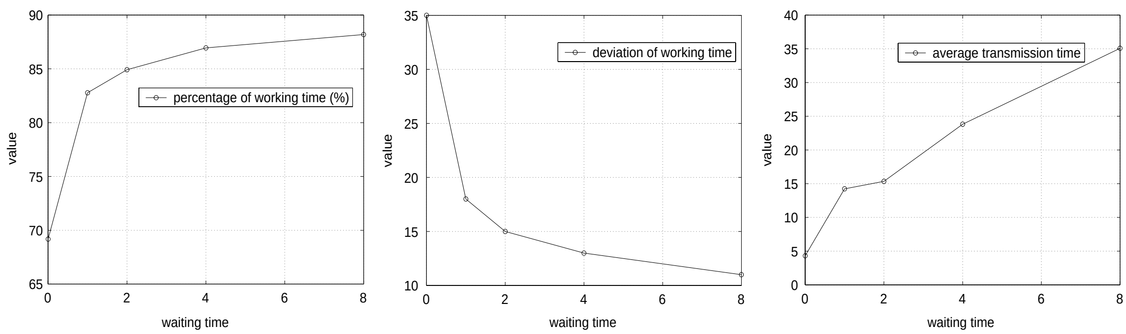

Fig. 5. The effect of varying the waiting time of hosts. The x-axis denotes the waiting time added to the hosts, while y-axis denotes the percentage of working time (left figure), the ratio of the standard deviation of the working times of the hosts and the average working time (middle figure), and the average transmission time of messages (right figure). The simulation was done with 20 hosts, a network space of 20×20, maximal host moving speed of 0.2, transmission range of 5.5, message arrival rate of 0.1, and a simulation time of 500. The basic waiting time vector was (0, 1.25, 0, 0.5, 0, 0.5, 0.125, 0, 0.125, 0.5, 0, 1.75, 0.625, 0.5, 0, 0.125, 1.625, 0.125, 1.125). For the x-axis, $1, 2, 3, 4$ correspond to waiting time multiplicative factors on the basic time vector. For example, in the 4th experiment, the waiting time of the first host is $0 \times 4 = 0$, the second is $1.25 \times 4 = 5$, ... etc. A value of 0 denotes Instantaneous Message Transmission.

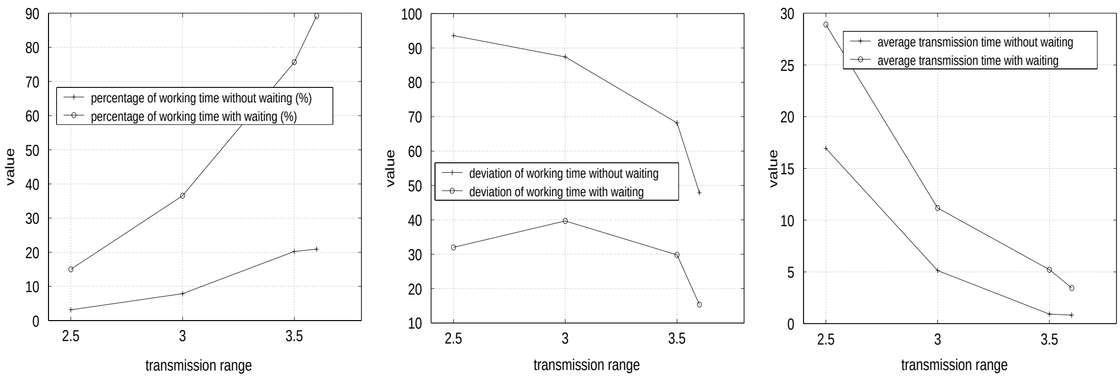

Fig. 6. The effect of varying the transmission range of hosts. The x-axis denotes the waiting time added to the hosts, while the y-axis denotes the percentage of working time (left figure), the ratio of the standard deviation of the working times of the hosts and the average working time (middle figure), and the average transmission time of messages (right figure). The simulations were done with 10 hosts, a network space of 10×10, the host moving speed of 0.2, message arrival rate of 2.0, and simulation time of 1000. The waiting time vector was $(2, 0, 20, 2, 0, 0, 20, 20, 20, 5)$.

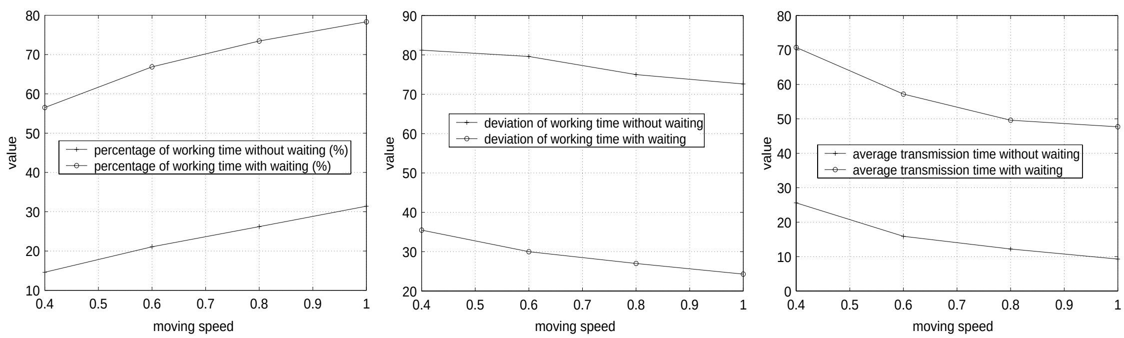

Fig. 7. The effect of varying the moving speed of hosts. The x-axis denotes the maximal moving speed of the hosts, while the y-axis denotes the percentage of working time (left figure), the ratio of the standard deviation of the working times of the hosts and the average working time (middle figure), and the average transmission time of messages (right figure). It was simulated with 10 hosts, a network space of 10×10, simulation time of 1000, message arrival rate of 2.0, transmission range of 3.0, and host moving speeds of 0.4, 0.6, 0.8, 1 separately. The waiting time vector was $(2, 0, 15, 2, 0, 0, 1, 20, 15, 0)$.

high and the distances of the host pairs are large compared with the host moving speeds. By analyzing the experimental data, we find that some hosts have less working time than other hosts. We call those hosts critical hosts. Those hosts are on many relaying paths in the network.

- When the message arrival rate is low, the distances of the host pairs are short as compared with the host moving speed of host. The algorithm gives a good solution according to the two criteria: percentage of time spent on message transmission and the message transmission time.

7. Conclusion

This paper describes how trajectory changes can be used to transmit messages in disconnected ad hoc networks. We present two algorithms. The first uses full knowledge of the motions of the mobile hosts within some uncertainty constrains. Location updates are employed in the second method where the full location knowledge is not available. These algorithms avoid the traditional waiting and retry methods, which do not cope well with disconnections.

We believe that this approach to communication is useful for the following two types of distributed applications: (1) When most of the network is connected (for example, a well-maintained framework for a sensor network), while some hosts are dispersed away from the framework, we do not have too many trajectory modifications to relay messages. (2) When the distance between two hosts is slightly larger than their transmission range, hosts need to move small distances to relay messages.

Acknowledgments

We thank Mahadev Satyanarayanan for suggesting this problem. We are very grateful to three anonymous reviewers for many insightful and helpful comments. This work has been supported in part by Department of Defense contract MURI F49620-97-1-0382 and DAR-

PA contract F30602-98-2-0107, ONR Grant N00014-01-1-0675, NSF CAREER award IRI-9624286, Honda corporation, and the Sloan foundation; we are grateful for this support.

References

[1] D. Kotz, R. Gray, S. Nog, D. Rus, S. Chawla, G. Cybenko, Agent Tcl: targeting the needs of mobile computers, IEEE Internet Comput. 1 (4) (July/August 1997) 58–67.

[2] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad-hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile computing, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–181.

[3] Z.J. Haas, A new routing protocol for the reconfigurable wireless network, in: Proceedings of the 1997 IEEE Sixth International Conference on Universal Personal Communications, ICUPC'97, San Diego, CA, October 1997, pp. 562–566.

[4] S. Murthy, J.J. Garcia-Luna-Aceves, An efficient routing protocol for wireless networks, ACM/Baltzer J. Mobile Networks Appl. MANET (1,2) (October 1996) 183–197.

[5] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, Comput. Commun. Rev. 24 (4) (October 1994) 234–244.

[6] E. Royer, C.-K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, in: IEEE Personal Communication, Vol. 6, April 1999, pp. 46–55.

[7] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of MobiCom 2000, Boston, MA, August 2000.

[8] K. Nakano, S. Olariu, Randomized leader election protocols in radio networks with no collision detection, in: Proceedings of 11th International Conference on Algorithms and Computation, ISAAC, Taipei, Taiwan, December 2000.

[9] Y.B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: Proceedings of ACM/IEEE MOBICOM'98, 1998, pp. 66–75.

[10] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networks 7 (6) (November 2001) 609–616.

[11] Q. Li, D. Rus, Sending messages to mobile users in disconnected ad-hoc wireless networks, in: MOBICOM, Boston, August 2000, pp. 44–55.

[12] I. Stojmenovic, X. Lin, Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks, IEEE Trans. Parallel Distributed Syst. 12 (10) (October 2001) 1023–1032.

[13] A. Boukerche, S. Rogers, GPS query optimization in mobile and wireless ad hoc networking, in: Sixth IEEE Symposium on Computers and Communications, Tunisia, Africa, July 2001, pp. 198–203.

[14] A. Boukerche, S.K. Das, Message traffic control capabilities of the r-dsdv protocol in mobile ad hoc networks, in: Proceedings of the ACM Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Rome, Italy, July 2001, pp. 105–114.

[15] A. Boukerche, S.K. Das, A. Fabbri, Analysis of randomized congestion control with dsdv routing in ad hoc wireless networks, J. Parallel Distributed Comput. 61 (2001) 967–995.

[16] M. Bui, S. Das, A.K. Datta, D.T. Nguyen, Randomized mobile agent based routing in wireless networks, Internat. J. Found. Comput. Sci. 12 (3) (2001) 365–384.

[17] I. Stojmenovic, X. Lin, Power aware localized routing in wireless networks, IEEE Trans. Parallel Distributed Syst. 12 (11) (November 2001) 1122–1133.

[18] Q. Li, J. Aslam, D. Rus, Online power-aware routing in wireless ad-hoc networks, in: MOBICOM, Rome, July 2001, pp. 97–107.

[19] B.R. Badrinath, T. Imielinski, Location management for networks with mobile users, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Publishers, Dordrecht, 1996, pp. 129–152.

[20] Z. Lei, C. Rose, Wireless subscriber mobility management using adaptive individual location areas for pcs systems, in: IEEE International Conference on Communications (ICC), Atlanta, GA, USA, June 1998.

[21] A. Boukerche, A. Fabbri, O. Yildiz, S.K. Das, Performance analysis of a parallel pcs network simulation, in: HiPC, Calcutta, India, 1999, pp. 189–196.

[22] J.-J. Fu, R.C.T. Lee, Minimum spanning trees of moving points in the plane, IEEE Trans. Comput. 40 (1) (January 1991) 113–118.

[23] J. Basch, L.J. Guibas, L. Zhang, Proximity problems on moving points, in: The 13th Symposium of Computational Geometry, 1997, pp. 344–351.