Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks

Qun Li
Department of Computer Science
Dartmouth College
Hanover, NH 03755
liqun@cs.dartmouth.edu

Daniela Rus
Department of Computer Science
Dartmouth College
Hanover, NH 03755
rus@cs.dartmouth.edu

ABSTRACT

An ad-hoc network is formed by a group of mobile hosts upon a wireless network interface. Previous research in this area has concentrated on routing algorithms which are designed for fully connected networks. The usual way to deal with a disconnected ad-hoc network is to let the mobile computer wait for network reconnection passively, which may lead to unacceptable transmission delays. In this paper, 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, . . . , and wireless routing algorithms. The simple try, timeout, sleep, retry loop can often fail particularly if the system does not happen to retry connection during a brief reconnection period. Waiting may be disastrous in some emergency cases. 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 have to 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. An ad-hoc network is formed by a group of mobile hosts upon a wireless local network interface. It is a temporary network without the aid of any established infrastructure or centralized administration. 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. This all-connected assumption is too strong for real world applications. 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.

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 change its trajectory based on the knowledge about other hosts trying to achieve the network connection actively. We believe that kind of active message transmission is useful for applications that require urgent message delivery and involve cars and robots, such as field operations or emergency relief.

In this paper, we explore the possibility of changing the trajectories of the hosts in a dynamic disconnected ad-hoc network to transmit messages among hosts. We show how the information about the motion of the destination host can be used to determine how the message can be sent by the cooperation of the intermediate hosts. We seek to minimize the trajectory modifications in order to transmit a message.

Two algorithms are studied in the rest of the paper. In the first algorithm, we assume the information about the motions and locations of hosts is known to all hosts, or can be estimated with some error parameters. The second algorithm does not assume that the movement of the hosts is known.

We envision an implementation of this approach using mobile agents ([7]). 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 the 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 in a remote or hazardous environment, the message routing program could suggest trajectory modifications for the team, while the individual robots can decide the ultimate host trajectories.

This paper introduces a new deployable ad-hoc network but does not address all the issues that are raised by this approach to communication. Many questions have to be answered in order to completely characterize the applications for which this approach to communication is suitable. What we accomplish here is to show that active message transmission by relay is a promising protocol for communication is ad-hoc wireless networks. We hope that this work will provide inspiration for more work 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 evaluates the performance. Section 7 concludes the paper.

2. RELATED WORK

We are inspired by recent progress in three areas: ad-hoc networks, personal communication systems, and mobile agents.

There has been a lot of work in the past on routing in ad-hoc networks [3, 2, 9, 11, 12, 13]. Routing algorithms have to cope with the typical limitations of wireless networks: high power consumption, low wireless bandwidth, and high error rates. The existing routing algorithms can be categorized by when the routes are determined. A pro-active algorithm probes the routes periodically. The message sent from one host to another can take the path which is already in the routing tables. A reactive algorithm triggers route searching when a host sends a message. The message can be sent after a route is discovered. Hybrid algorithms combine the pro-active and reactive methodologies. The algorithms are generally based on the link-state information between the neighboring hosts. Some other interesting algorithms make use of the GPS[1] location information to aid the route discovery, for example the location aided routing (LAR) [6]. LAR limits the search for a route to the "request zone", determined based on the expected location of the destination node at the time of route discovery. By using the GPS location information, a host can make decisions on whether it will forward the message to its neighbors or discard it.

Another related area is Personal Communication System (PCS) [14, 8] location management. PCS enables people to communicate independent of their locations. The system needs a location management mechanism to locate the mobile users, which maps subscriber numbers to the current location of the requested users. This is implemented as a central database for mapping subscriber numbers to locations. Two operations are used for database maintenance and retrieval. The update operation informs the system database regarding changes in mobile user locations. The search operation locates the mobile user by searching the database and paging the user in the network.

Many location management methods have been proposed. They differ in the way the location database is organized. Most location management techniques use a combination of updating and finding in an effort to select the best tradeoff between the update overhead and the delay incurred searching. Specifically, updates are not usually sent every time an 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.

Our proposed approach to message transmission in wireless networks extends the concept of an "active message" introduced by ([10]). An active message is a communication strategy based on mobile agents, which has been proposed to handle the network disconnection in the wireless network. The idea is to use light weight mobile agents as the message carriers. The active message is capable of jumping from one node to another according to the message path which is defined by a routing algorithm. The message is forwarded hop by hop with the possibility that it may reside on some intermediate node due to the loss of a network link.

3. MESSAGE TRANSMISSION UNDER FULL KNOWLEDGE OF THE HOST LOCATIONS AND TRAJECTORIES

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

[1]GPS stands for global position system

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.

3.1 Communicating Multiple Messages

We assume a scenario in which a set of hosts move according to pre-specified trajectories and the maximal possible speeds of hosts is high with respect to the distance between hosts. In such situations, the time for a host to approach another host is quite short, and it doesn't affect the motion estimation by any other hosts. That is, the time spent by a host deviating from the original trajectory is negligible.

Each host in the system has a task to carry out, that 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 (Figure 1).

For each host h_i in the system
pursue investigation while waiting to receive messages.
generate message when needing to communicate.
if a message m_j is received
  if the recipient of m_j is h_i process m_j
  else if the recipient of m_j is h_k compute
    optimal_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.

Figure 1: Algorithm 1: An algorithm for the behavior of each host h_i in an ad-hoc network that uses relays to communicate messages.

We continue by focusing on the technical details that make this communication scheme realistic. We first discuss how an individual message is routed in a mobile ad-hoc network. Clearly, if the routing of a message requires host trajectory modifications, this has the potential for interfering with the concomitant transmission of other messages. There are two main issues: (1) how can a host be found if the host is gone on a relay task? and (2) what happens if a host has to relay multiple messages in different directions? Depending on the frequency of the messages, the speeds of the hosts, and their transmission range, these may be big issues with our model, or no issue at all. The answer really depends on the application scenario. In this paper we quantify trade-offs between these parameters in the hope that our results could be used to decide whether an application is suited to this approach to communication or not. We believe that typical field operation missions (such as the recent mission in Kosovo) and emergency relief (such as in an area where the infrastructure has been destroyed by a natural disaster) where the teams move relatively close to one another and communication is done purely on the ad-hoc network, our proposed approach is valuable. An important open problem here is under what conditions can we offer performance guarantees for communicating multiple messages in parallel. One possible solution is to use a network flow algorithm to model this case[2].

It is clear that deadlock may occur in some situations, and characterizing all the cases when this may occur is an open problem. For now, we observe that if a all the hosts in the system decide to approach each other at the same time, in a circular fashion, we have deadlock. To cope with deadlock, one possible solution is to employ message discard and retransmission. Hosts could wait for a fixed amount of time at the estimated location of the message recipient. If communication is not possible during this fixed amount of time, the host could return to its original trajectory and retry transmission later (or discard the message.) It is clear that this is a complicated protocol and for certain motion patterns it is not feasible. In this paper, we attempt to introduce the paradigm of message transmission using relay and describe our first results in characterizing this protocol. Much work is left to be done in order to understand clearly for what type of applications is this a suitable approach. For example, using more precise movement descriptions (such as communication with a walkie-talkie to update the location for all neighboring hosts) we envision a more sophisticated scheme in which message transmissions and trajectory changes can be computed by considering all the messages being sent.

3.2 Communicating One Message

In this section we assume that all the hosts' motion descriptions are known.

Suppose h_1, h_2, h_3, h_4 are four mobile hosts in an ad-hoc network (see Figure 2) with known motions at dispatch time. If h_1 wants to send a message to h_4 and h_4 is not within its transmission range, h_1 needs to get closer to h_4. 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 approach h_4 by h_1 itself without any help from other hosts. 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 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 without any intermediate host (we describe this algorithm in Section 3.2.1). The shortest trajectory (say to host h_2) may provide a faster way of reaching the transmis-

[2]The following reduction maps the multiple message relay protocol to multi-commodity flow. Let G be the host graph, where each host corresponds to a vertex in G. Suppose we can estimate the time it takes a host to approach another host. The time required to send a message between two hosts can then be used as the weight of the edge connecting the two hosts in the host graph. Next, we restrict the amount of time spent transmitting a message in a time unit to favor the activity of the host towards the global task. This can be used to constrain the capacity of each edge in the graph.

sion range of the other hosts. The shortest trajectories can thus be computed incrementally using (possibly) more and more intermediate hosts. The Optimal Relay Path can be

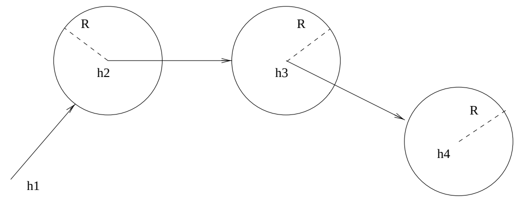

Figure 2: h_1 sends a message to h_4 by way of intermediate hosts h_2 and h_3. Disks corresponds to the transmission range of hosts and arrows show approach trajectories to relay messages.

formalized under the following assumptions:

1. Two hosts can communicate with each other within the range of 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 messenger to send the message to h_4. For example, in Figure 2, 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 Optimal Relay Path algorithm, we introduce the following terminology.

DEFINITION 1. *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 before it receives the message of other hosts and prepares to relay it.*

DEFINITION 2. **A moving path** *from A to B is a sequence of hosts, $h_0, h_2, ..., h_k$ (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 Figure 2, $h_1h_2h_3h_4$ with the moving strategy of each host is a moving path from h_1 to h_4.*

DEFINITION 3. **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.*

Figure 11 describes the Optimal Relay Path algorithm, which determines the shortest path to the destination of the message. The algorithm computes the time of 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 haven't been processed, and marking the host ready. Then it updates the current minimal time from h_0 to all hosts that are not ready. The running time of the algorithm is $O(n^2t)$ where t is the running time of algorithm $OptimalTrajectory$.

This algorithm can be employed if the application has the following characteristics: if the maximal possible speed of the hosts in the system is larger than the moving speed of the host which is being approached, the message can be sent successfully given the moving descriptions of the hosts.

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. More precisely, by solving the equations, the velocity of host h_m and the approaching time can be obtained.

$$|\vec{P_j(t)} - (\vec{P_m(t_0)} + \vec{v} \cdot (t - t_0))| \leq R \quad (1)$$

$$\frac{\vec{P_j(t)} - \vec{P_m(t_0)}}{|\vec{P_j(t)} - \vec{P_m(t_0)}|} = \frac{\vec{v}}{|\vec{v}|} \quad (2)$$

Equation 1 gives the condition for host h_j to be in the transmission range of host h_m when h_m approaches h_j. Equation 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. Since the maximal speed of the host is known, the equations have two unknown variables: the approaching time point t, and the moving direction θ.

Equations 1 and 2 lead to an algorithm for computing optimal trajectories. The solution depends on the movement of the hosts in the system. We examine two cases.

Case 1: All hosts are static before they receive the message and are required to move to relay message. In (Figure 3, left), the initial positions of hosts $h_1, h_2,$ *and* h_3 are A, B, C. Suppose h_1 wants to send a message to a host which is within the range of h_1, such as h_2, h_1 can send the message immediately. If the host is out of the transmission range of h_1, such as h_3, h_1 should move towards h_3 for $|AC| - R$ where

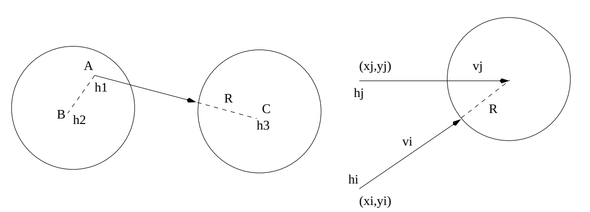

Figure 3: Two examples of approaching hosts, using the host movement information. The disks represent transmission range. In the left figure, all hosts are static. h_1 can send messages to h_2 directly. But it should move toward h_3 for communication. The right figure shows optimal trajectory of h_i starting from (x_i, y_i) to approach h_j, while h_j is moving to right starting from (x_j, y_j).

Input: initial time when host h_0 begins to send a message. and 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 $h_1, h_2, \cdots, h_n$.

1. Compute the optimal trajectory for host h_0 to reach all other hosts directly, record the earliest time point $t[i]$ for h_i.
2. Choose the unmarked host h_i with the least $t[i]$, mark h_i.
3. Compute the optimal trajectory for host h_0 to reach unmarked hosts, such as, h_j by way of h_i. If the time point computed is less than $t[i]$, update $t[i]$.
4. Goto 2 until all hosts have been marked.

Figure 4: Algorithm 2: the Optimal Relay Path to all hosts in the system.

$|AC|$ is the distance between A and C. The running time of algorithm *OptimalTrajectory* is $O(1)$.

Case 2: For any host h_j, it keeps its fixed velocity v_j before it receives the message from another host (Figure 3, right). Suppose host h_i wants to move to approach h_j. Initially, if host h_j is within the transmission range of h_i, h_i can send the message directly. Otherwise, the movement of host h_i can be described by the following equation.

$$(\vec{v_j} - \vec{v_i}) \cdot t + (x_j - x_i, y_j - y_i) = R \cdot \frac{\vec{v_i}}{|\vec{v_i}|}$$

where (x_i, y_i) and (x_j, y_j) are the initial position of node i and j. The running time of the *OptimalTrajectory* algorithm is $O(1)$.

Similar equations can be derived when the trajectory is given as a function of acceleration and velocity; we omit these computations as they are quite involved.

4. MESSAGE TRANSMISSION UNDER LOCATION ERROR

An important property of the Optimal Relay Path algorithm (see Figure 11) is that it works even if the location of 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 (see Figure 11) 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 how much the hosts have to move in order to relay a message from the originator to the destination. In other words, we compute the sum of distances traveled by each host involved in the transmission of one message. The exact computation of the traveled distance does hot hold in this case because the location of hosts is known only approximately and some 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*. This restriction is realistic when the moving speeds of the hosts are relatively slow. The upper bound for the total movement necessary to relay a message is given by the following result:

THEOREM 1. *The length of the moving path computed by the Optimal Relay Path algorithm in Figure 11 is at most $(4n-5)r$ more than the length of the optimal path, where n is number of the hosts in the system.*

Proof sketch. In order to compare the path we get with location error with the optimal path, we take an imaginary system in which all hosts are static as a reference. By comparing the imaginery system with the path obtained in our algorithm and the optimal path, we can get the difference between the length of the path in our algorithm and that of the optimal path. The proof of the theorem is based on two lemmas which can be found in the Appendix.

In order to decrease the error of the location estimation, the previous algorithm can be refined by adding the exchange of the 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 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 an 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 at the most recent time point. In this manner, the up-to-date motion information of the hosts will be propagated.

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

Figure 5 gives simulation data for how the location error affects the performance of the system. With a larger location error (and thus a larger uncertainty), a host needs more time to approach another one.

5. COMMUNICATION WITHOUT FULL KNOWLEDGE OF THE HOST MOTIONS

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, and the host's motion information has not been propagated for a long time. When the error is larger than the transmission range, tracing

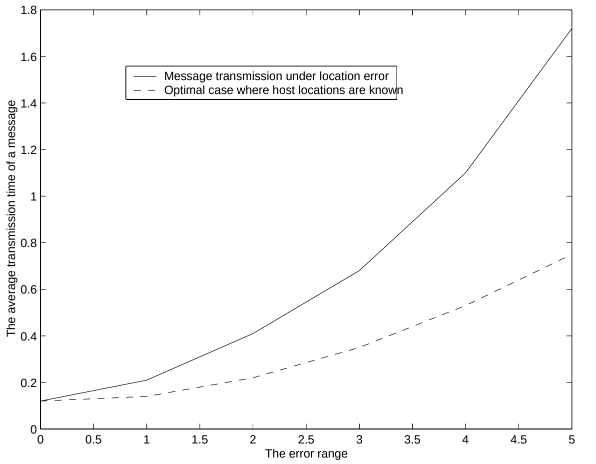

Figure 5: This graph shows the comparison between message transmission under location errors and the optimal case in which all host locations are known. The x-axis denotes the error range of the simulations, that is, the maximal error of the guessed the locations. The y-axis denotes the average time 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 5, and a message arrival rate 0.2, 0.1, and 0.02 for each host (we average the transmission time). The simulation was run for 10^5 seconds. For the x-axis, 0 means the hosts are static at their own position, and 1 means a host can be anywhere in a circle with radius 1 centered on its guessed location.

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 *a priori* 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 should a host send out information about its location update; (2) to whom should the host send out this information; and (3) how should the host send out this information. In this section we present solutions to (1) and (2) that 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 are the hosts that keep track of its location. The host location update should occur when the host leaves its current scope. For example, consider a fully connected network with one mobile host on its periphery. The network can keep track of the mobile host if the mobile host communicates its current location periodically.

We model the communication problem in a mobile ad-hoc network as a minimum spanning tree. Let G be a weighted

---

Notation:
$\mathbf{t_i}$: 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 $h_1, h_2, \cdots, h_k$ that are adjacent to h_0 in the minimum spanning tree.
Compute the optimal radius r_i between h_0 and h_i.
If $|(x_t, y_t) - (x_{t_i} + v_x(t - t_i), y_{t_i} + v_y(t - t_i))| \geq r$, move to h_i to update its location ($t_i = t$ and $(x_{t_i}, y_{t_i}) = (x_t, y_t)$).
If there is message exchange between h_0 and h_i, update the t_i and (x_{t_i}, y_{t_i}) to the current time and location.

Figure 6: This figure shows pseudo-code for the algorithm used for location updates when the hosts do not know a priori their moving paths.

graph whose vertices correspond to the hosts in the system. 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 only inform 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 tradeoff in the next section. Figure 6 gives the algorithm for the location update in this communication method.

5.1 The Communication between Two Hosts

In this section we analyze the trade-off between scope and update frequency, by considering the error in a host's estimation about the location of another host, in 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 tree can be done using previous algorithms for dynamically constructing a minimum spanning trees [1, 4].

Suppose there are two hosts which communicate with each other, and are out of transmission range of each other. There are two types of communication: one is true message communication, the other is location update. Each host has its own task to carry out which may require it to move. 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 they also 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_i will have to update its location more frequently, which means that h_j will travel a lot for this purpose. If the scope size is large, h_i has to do fewer location updates, but h_j has a less good approximation for where h_j is so h_j 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 radius for the scope indicates 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 this time-varying behavior of a mobile host is the Brownian motion with a drift process [5]. The two dimensional Brownian motion with a drift process can be described by the distribution:

$$p_{xy}(x, y|x_0, y_0, t) = \frac{1}{2\pi\sqrt{D_x D_y}(t - t_0)} \cdot$$

$$\exp\left(\frac{-[(x - x_0) - v_x(t - t_0)]^2}{2D_x(t - t_0)} + \frac{-[(y - y_0) - v_y(t - t_0)]^2}{2D_y(t - t_0)}\right) \quad (3)$$

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 Equation 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: $r(t) = \sqrt{2D(t - t_0)\ln(\frac{1}{1-\gamma})}$ And 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 ($l \geq R$), 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 of the hosts. We assume that the maximal possible speed of a host is quite large compared with the host's general moving speed. Thus, the host doesn't need to consider the effect of the message transmission or location updating time.

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

$$Y = (\lambda + \frac{1}{t_r})(l - (R - 2r)) \quad (4)$$

where $l - (R - 2r)$ is the maximal distance for the host travels to approach another host by the analysis in Lemma 4. We want to minimize Y subject to $r \leq R$. The following theorem shows that Y can only obtain its minimal value at some roots of a cubic equation or at R.

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

PROOF. As we know

$$t_r = \frac{r^2}{2D \cdot \ln(\frac{1}{1-\gamma})}$$

Let $P = 2D\ln(\frac{1}{1-\gamma})$, we have $Y = 2\lambda r + \frac{2P}{r} + \frac{P(l-R)}{r^2} + \lambda(l - R)$. Take derivative of r to the both sides of the equation, and set the derivative to zero, we get a cubic equation. Solve the equation, let

$$d = \sqrt{\frac{P^3}{27\lambda^3}}, \text{ and } \theta = \cos^{-1}(\frac{3}{2} \cdot (l - R) \cdot \sqrt{\frac{3\lambda}{P}})$$

Thus the minimal value of Y can only be obtained when r is $2 \cdot (\frac{P(l-R)}{2\lambda})^{\frac{1}{3}}$, $2 \cdot d^{\frac{1}{3}} \cdot \cos\frac{\theta}{3}$, or R. □

Since there are three possible places for attaining the minimum value for r, we would like to experimentally study when exactly the optimum happens. Figure 7 shows the solution for the optimum radius (defined by Equation 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.

Figure 7 (left) describes the change of the optimal radius as m grows. The curves are plotted with for $k - 1 = 8, 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128$ and $\gamma = 95\%$. Figure 7 (right) shows the optimal radius change with the change of the $k - 1$. It includes five curves with $m = 1/8, 1/16, 1/32, 1/64, 1/128$. Except for the $m = 1/8$ curve, 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, it takes $2 \cdot (\frac{P(l-R)}{2\lambda})^{\frac{1}{3}}$. With k increases, it takes $2 \cdot d^{\frac{1}{3}} \cdot \cos\frac{\theta}{3}$. It takes R when k is quite large.

The length of trips traveled by the hosts is determined by the length of a single trip and the the number of trips. Figure 7 shows that the bigger k is, the longer the optimal radius is. That is because for a large k (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 of m affects the 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 update message transmission 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.

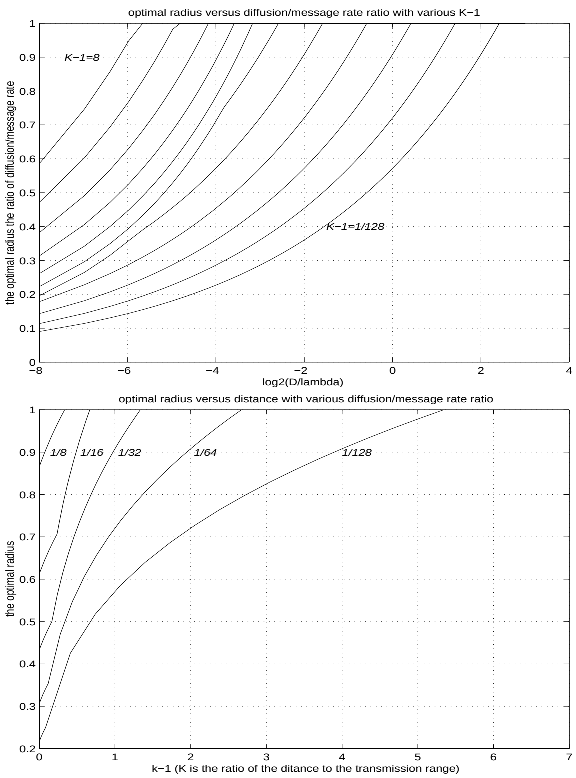

Figure 7: This figure shows the optimal radius of the scope for the hosts. The left figure shows the dependency of this radius (represented by the y-axis) on the ratio $\frac{D}{\lambda}$. Each curve is drawn for different values of k, the ratio defined by distance between two hosts, divided by the transmission 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 $\frac{D}{\lambda}$

6. EXPERIMENTS IN SIMULATION

In this section we explore in simulation application properties when the distributed application communicates messages using the trajectory modification relay scheme. Our goal is to collect empirical data that characterizes how hosts divide their times between relaying messages and executing their own task. In the future, we hope to develop analytical models that will predict how much time a host spends relaying messages; such a model could be used to establish tradeoffs between our proposed scheme that guarantees message arrival and other possible approaches to communication.

We have developed a simulation to evaluate the performance of our algorithms. In these experiments, we focus on evaluating how the 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 times and the average working time, and the average transmission of a message. The first metric denotes how much time the hosts spend on its own work instead of message transmission. The second metric is to evaluate how much the workloads of the hosts are balanced. And the third one 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 (Figure 11) 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, when a host receives a message, instead of transmit it immediately, it delays its transmission after working for some time (we call it waiting time in the following), which is a parameter to the experiment. In the following, we denote the waiting time for each host using the waiting time vector each component of which gives the corresponding amount of waiting time for each host. 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 the host in the waiting period are sent to the next host as a group if their next hosts are the same.

Figures 8, 9, and 10 show the data we compiled from these experiments.

Figure 8 demonstrates the effect of varying the waiting times of hosts. Typically, the working time increases with larger waiting times. With a larger waiting time, more messages are accumulated at a host, thus some messages may be sent together. The average message transmission time also increases with the increase of waiting time. For the metrics of percentage of working time and ratio between deviation and average working time, Delayed Message Transmission is always better than Instantaneous Message Transmission. We also observe 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. increases little after some point of the working time increase.

Figure 9 shows the comparison between Instantaneous Message Transmission and Delayed Message Transmission while the transmission range is changed. As the transmission

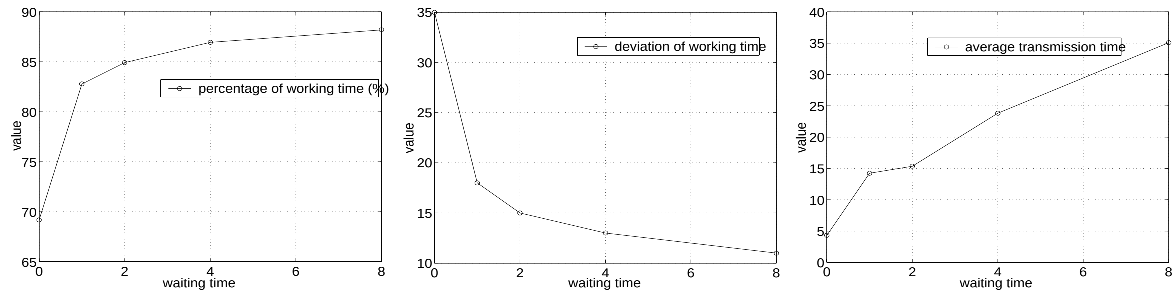

Figure 8: The effect of varying the waiting time of hosts. The x-axis denotes the waiting time added to the hosts, while y-axis denotes the percentage of working time (left figure), the ratio of the standard deviation of the working times of the hosts and the average working time (center figure), and the average transmission time of messages (right figure). The simulation was done with 20 hosts, network space is $20 * 20$, maximal moving speed of host 0.2, transmission range 5.5, message arrival rate 0.1, simulation time 500. The basic waiting time vector is (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). And in x-axis,$1, 2, 3, 4$ mean the waiting time vector in each case is that times of the basic waiting time vector. For example, in the experiment of 4, the waiting time of the first host is $0 * 4 = 0$, the second is $1.25 * 4 = 5$, $\cdots$ and so on. And 0 means Instantaneous Message Transmission.

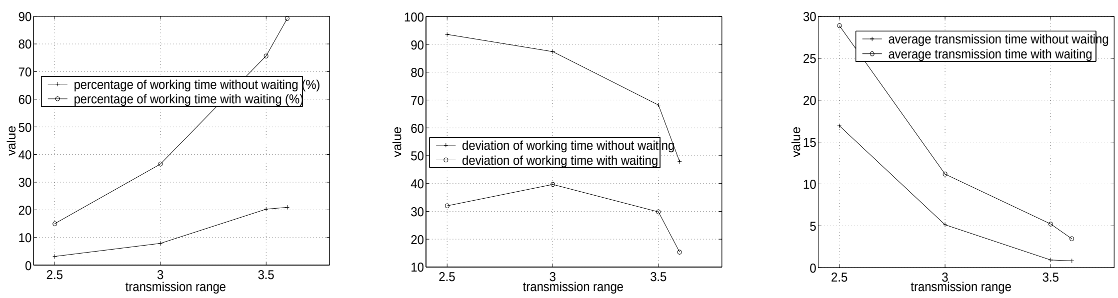

Figure 9: The effect of varying the transmission range of hosts. The x-axis denotes the waiting time added to the hosts, while y-axis denotes the percentage of working time (left figure), the ratio of the standard deviation of the working times of the hosts and the average working time (center figure), and the average transmission time of messages (right figure). The simulations were done with 10 hosts, a network space 10*10, the moving speed of host 0.2, message arrival rate 2.0, and simulation time 1000. The waiting time vector is (2,0,20,2,0,0,20,20,20,5).

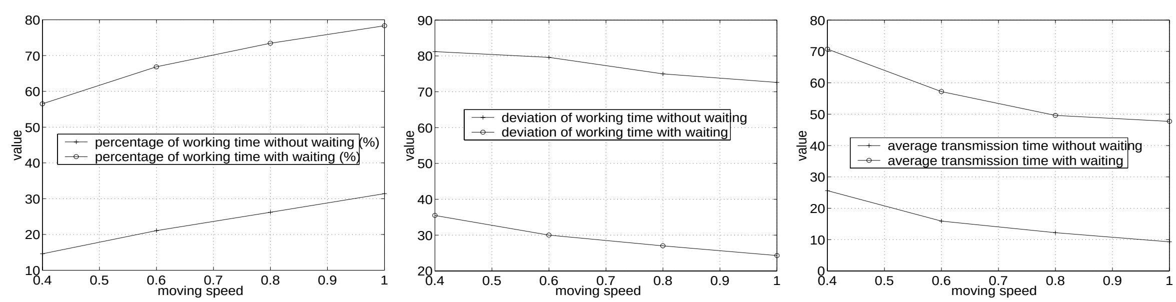

Figure 10: The effect of varying the moving speed of hosts. The x-axis denotes the maximal moving speed of the hosts, while y-axis denotes the percentage of working time (left figure), the ratio of the standard deviation of the working times of the hosts and the average working time (center figure),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

range increases, the working time increases, and the average message transmission time decreases. The larger transmission range contributes to the shorter travel path for a host, in turn affects the message transmission time and working time. As for the percentage of working time and ratio of deviation and average working time, Delayed Message Transmission does much better than Instantaneous Message Transmission in every case.

Figure 10 shows the influence of the various maximal speed of the hosts on the performance. It is obvious that a larger speed improves the 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 high and the distances of the host pairs are large compared with the moving speed of host. By analyzing the experimental data, we find some hosts have less working time than other hosts. We call those hosts critical hosts. Those hosts are on many paths of the other hosts in the network. Besides, they are also some distance away from other hosts.
- When the message arrival rate is low, and the distances of the host pairs are short as compared with the moving speed of host. The algorithm gives a good solution according to the two criteria: percentage of time spent on message transmission an the message transmission time.

7. SUMMARY

This paper describes how the trajectory change can be used to transmit messages in disconnected ad-hoc networks. We present two methods to solve the problem. The first uses the full knowledge of the motions of the mobile hosts, or with some limited errors. Location update is employed in the second method where the full knowledge is unknown. These algorithms avoid the traditional waiting and retry method, which is intolerable in some emergency case.

We believe that this apprach to communication is useful for the following two types of distributed applications. (1) In the case when most of the network is connected (such as 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) In the case when the distance between two hosts is slightly larger than the transmission range, hosts need to move small distances to relay messages.

It is clear that much work remains to be done in order to fully understand this model of communication. Our most immediate goal is to implement a physical experiment on our platform of 24 laptops equipped with wireless modems for the urban warfare application we developed for our MURI project. In this application, soldiers equipped with wireless laptops patrol around an area and cooperate to identify a villain and secure a building. We have already demonstrated the use of mobile agents for providing communication in such a situation but the messages did not always reach their targets in due time, because often the hosts were out of range. A natural extension would be to incorporate the message relay idea to guarantee the delivery of messages. We hope to do this experiment in the next few months.

Acknowledgements

We thank Mahadev Satyanarayanan for suggesting this problem. We are very grateful to three anonymous reviewers for many insightful and helpful comments. This work bas been supported in part by Department of Defense contract MURI F49620-97-1-0382 and DARPA contract F30602-98-2-0107, ONR grant N00014-95-1-1204, NSF CAREER award IRI-9624286, Honda corporation, and the Sloan foundation; we are grateful for this support.

8. REFERENCES

[1] Jyh-Jong Fu and R. C. T. Lee. Minimum spanning trees of moving points in the plane. *IEEE Transactions on Computer*, 40(1):113–118, Janurary 1991.

[2] Z. J. Haas. A new routing protocol for the reconfigurable wireless network. In *Proceedings of the 1997 IEEE 6th International Conference on Universal Personal Communications, ICUPC'97*, pages 562 –566, San Diego, CA, October 1997.

[3] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad-hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153 –181. Kluwer Academic Publishers, 1996.

[4] Leonidas J. Guibas Julien Basch and Li Zhang. Proximity problems on moving points. In *the 13th Symposium of Computational Geometry*, pages 344–351, 1997.

[5] S. Karlin and H. M. Taylor. *A First Course in Stochastic Processes*. Academic Press, New York, 2nd edition, 1975.

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

[7] David Kotz, Robert Gray, Saurab Nog, Daniela Rus, Sumit Chawla, and George Cybenko. Agent Tcl: Targeting the needs of mobile computers. *IEEE Internet Computing*, 1(4):58–67, July/August 1997.

[8] Zhuyu Lei and Christopher 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.

[9] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. In *ACM/Baltzer Journal on Mobile Networks and Applications*, volume MANET, pages 183 –197, October 1996.

[10] C. Okino and G. Cybenko. Modeling and analysis of active messages in volatile networks. In *Proceedings of the 37th Allerton Conference on Communication, Control, Computing*, Monticello, IL, September 1999.

[11] V. Park and M. S. Corson. A highly adaptive distributed algorithm for mobile wireless networks. In *Proceedings of INFOCOM'97*, Kobe, Japan, April 1997.

[12] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Computer Communication review*, volume 24, pages 234 –244, October 1994.

[13] Elizabeth Royer and C-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. In *IEEE Personal Communication*, volume 6, pages 46 – 55, April 1999.

[14] T.Imielinski and H. Korth. Location management for networks with mobile users. In *Mobile Computing*, pages 129–152. Kluwer Publishers, 1996.

9. APPENDIX

9.1 Proof of the optimality of the optimal relay path algorithm

We want to prove 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 1. *In an environment with only two hosts, if the motion of host h_j is predictable, thus 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. Suppose *OptimalTrajectory* is the algorithm to compute the optimal moving path from h_i to h_j at time point t_0 given the motion function of h_j and the motion description of h_i. Let $(t, \theta) = OptimalTrajectory(P_i(t_0), P_j(t), t_0)$ where $P_j(t)$ is the motion function of host h_j, $P_i(t_0)$ is the position of host h_i at time point t_0, t is the time elapsed for h_i to approach h_j and send the message to h_j, t_0 is the initial time point, and θ is the moving direction of h_i to approach h_j. *OptimalTrajectory* may be complex. However, due to the predictability of h_j's movement, *OptimalTrajectory* does exist. □

LEMMA 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 of h_0 from another host, it should move according to the path given by algorithm OptimalTrajectory.*

PROOF. Otherwise, we can replace the path of h_{b-1} with the path given by the algorithm *OptimalTrajectory*. And we get a better moving path, which is a contradiction. □

LEMMA 3. *If $h_0, h_1, ..., h_b$ is an optimal moving path from h_0 to h_b, then $h_0, h_1, ..., h_{b-1}$ must be an optimal path from h_0 to h_{b-1}.*

Notations:
$\mathbf{P_j(t)}$: the function describing the trajectory of host j.
Ready[j]: a flag for host h_j denoting if the shortest path to h_j has been found.
t[j]: the earliest time (currently known) for the message to get host h_j.
P[j]: the position of host h_j at time $t[j]$.
m: h_m is the most recent host whose optimal path has been found.
$(\mathbf{T}, \theta) = \mathbf{OptimalTrajectory}(\mathbf{P_m(t[m]), P_j(t), t[m]})$: The function calls the algorithm which gives the optimal trajectory approaching h_j by h_m starting from $t[m]$. The function returns the time required and the moving direction of h_m.
Input:
$\mathbf{t_0}$: initial time when host h_0 begins to send a message.
$\mathbf{P_i[t]}$: the moving function of host h_i, which gives the position of h_i at time t.
Output:
Path[i]: the optimal moving path from host h_0 to all other hosts $h_1, h_2, \cdots, h_n$.

```
/* initialization */
t[0] = t0;
Path[0] = {};
Ready[0] = 1;
for i = 1 to n − 1
begin
  (T, θ) = OptimalTrajectory(P0(t0), Pi(t), t0);
  /* the direct path from host 0 */
  t[i] = t0 + T;
  Ready[i] = 0;
  /* to host i without other intermediate hosts */
end
/* main body */
for i = 1 to n − 1
begin
  Let t[m] be the minimum among all t[j] where
    Ready[j] = 0;
  Ready[m] = 1;
  /* optimal path for host m has been found */
  for all nodes j where Ready[j] = 0
  begin
    (T, θ) = OptimalTrajectory(Pm(t[m]), Pj(t), t[m]);
    if (T + t[m] < t[j]) /* update the path for host j */
    begin
      t[j] = T + t[m];
      /* if the path via m is a better one */
      Path[j] = Path[m] ∪ {(m, θ)};
    end
  end
end
```

Figure 11: Algorithm 2: the Optimal Relay Path to all hosts in the system.

Proof. Suppose $P_n : h_0, h_1, ..., h_{b-1}$ is not optimal. Let the optimal moving path from h_0 to h_{b-1} be P_o. In P_n, h_{n-1} receives the message of h_0 at time point t_n, and in P_o, it receives at t_o. Thus, $t_n > t_o$. Before the time point t_o, the movements of h_{b-1} in P_n and P_o are the same. But after t_o, h_{b-1} begins to approach h_b in P_o, while in P_n, h_{b-1} starts to approach h_b after t_n. Note that h_b will not change its moving function before h_{b-1} is within its range of communication and sends message to it. So, P_o can give a better moving path to send message to h_b. It is a contradiction. □

Corollary In an optimal path from $h_0, h_1, ..., h_b$, the sub-path $h_0, h_1, ..., h_i$ (where $0 \leq i \leq b$) is an optimal moving path from h_0 to h_i, and the movement of h_i after it receives the message of h_0 can be achieved by applying the algorithm $OptimalTrajectory$. Thus the optimal moving path from h_0 to h_b can be constructed incrementally (such as, applying dynamic programming as we are doing now).

Theorem 3. *The Optimal Relay Path algorithm (Figure 11) gives the optimal moving paths from host h_0 to all other hosts.*

Proof. Prove by mathematical induction.

Let $h_0, h_1, ..., h_{n-1}$ be the sequence in the order of hosts whose h_i] is marked.

1. Initially, $Ready[h_0]$ is marked. The optimal moving path from h_0 to h_0 is itself.

2. Suppose after Ready[h_i] is marked, the algorithm gives the optimal moving paths from h_0 to h_1, h_2,...,h_{i-1}, and h_i.

3. Just after Ready[h_{i+1}] is marked. We get a moving path with the minimal time among all the moving paths (from h_0 to h_{i+1}) which only consists of the marked hosts.

a. It is optimal moving path if we only consider the marked hosts.

By the above corollary, the optimal moving path from h_0 to h_b can be divided into two parts. One is the optimal moving path from h_0 to h_{b-1}, another is the optimal path in which h_{b-1} moves to approach h_b and send message to h_b. By induction, we have got all the optimal moving paths from h_0 to h_1, h_2,..., and h_{b-1}, thus, by the analyzing the algorithm (the essence of the algorithm is to enumerate all possible paths to reach h_b via $h_0, h_1, ..., h_{b-1}$), we get the path with the minimal time, which must be the optimal moving path among all the marked hosts.

b. It is the optimal moving path from h_0 to h_{i+1} in the entire system.

If it is not the optimal moving path, the optimal moving path must consist of some unmarked hosts. Let h_k be the first such host in the path. So in the optimal moving path, the time of sub-path h_0 to h_k, which only consists of the hosts which has been Ready, is less than the time of the moving path from h_0 to h_{i+1}. So h_{i+1} doesn't have the minimal time at that time. Our algorithm shouldn't choose h_{i+1} and mark it. We get a contradiction. □

9.2 Proof of the theorem with location error

Let $E(f_1(t), f_2(t), \cdots, f_n(t), R, r)$ be an environment in which the estimated moving descriptions of hosts $h_1, h_2, \cdots, h_n$ are $f_1(t), f_2(t), \cdots, f_n(t)$, the transmission range is R, and the maximal location error is $r \leq R$. When $r = 0$, there is no location error with the estimated descriptions of the hosts. If a host h_i's location is estimated at (x_i, y_i), the actual location of the host h_i is possibly at anywhere in the circle of radius r centered at (x_i, y_i). We assume $r \leq R$ because the mobile host is hard to be found if the error is too big.

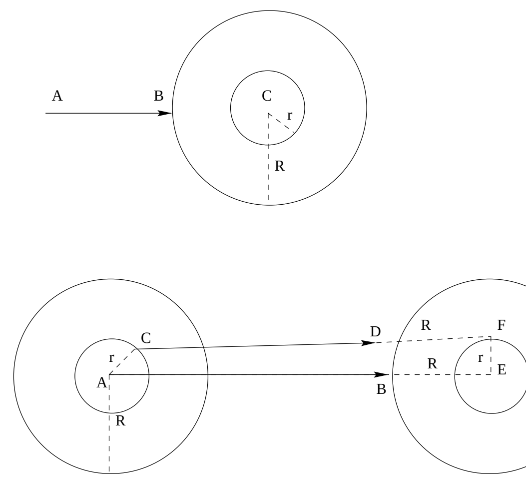

Figure 12: Two hosts in an environment with location error. The bigger disks represent transmission range (R), and the smaller ones are moving scope (with maximal error r). In the left figure, h_j starts from A to approach h_i. The right figure compares CD, the optimal path of h_j and AB, the path computed by our algorithm 2 using the estimated locations.

Next, we want to analyze the algorithm in an environment in which any host h_i in the system never moves beyond the range of a circle with center O_i and radius r (that is, $\vec{d_i}(t) = O_i$).

Theorem 4. *In an environment with location error, suppose our estimated moving description of host h_i is 'static at O_i'. Then the sum of the length of the moving path computed in our Optimal Relay Path algorithm 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.*

Lemma 4. *Suppose our estimated moving description of host h_i is 'static at O_i'. Algorithm 2 computes the moving path $P_1 : (h_1 h_2 \cdots h_k)$ in $E((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n), R, 0)$, say E_1. Let the length of P_1 is s_1. So in $E((x_1, y_1), (x_2, y_2), \cdots,$ (x_n, y_n), R, $r)$, say E_2, we have a moving path $P_2 : h_1 h_2 \cdots h_k$ which has the moving length s_2. We have $s_2 - s_1 \leq (2k - 3)r \leq (2n-3)r$*

PROOF. As Figure 12 (upper) illustrates, h_j starts from A to approach h_i. 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. So h_j can go further with the distance at most r (we call it an extra move) before it can send the message to h_i. After h_i receiving the message from h_j, it begins to approach the next host. h_i can first go back to C (another extra move) before it approaches the next host. In this period, the moving length of host h_i is r more than that of predictable environment, so is the moving length of host h_j. From h_1 to h_k, we have $2(k-1)-1$ extra moves in the system with estimation error, thus we have $s_2 - s_1 \leq (2k-3)r \leq (2n-3)r$. □

LEMMA 5. *For a moving path $P_3 : h_1 h_2 \cdots h_k$ in $E(d_1(t), d_2(t), \cdots, d_n(t), R, 0)$ (say E_3) where $|d_i(t) - (x_i, y_i)| \leq r$. Let s_3 be the length of P, then we have $s_1 - s_3 \leq (2n-2)r$.*

PROOF. In Figure 12 right, the bigger circles give the transmission range, while the smaller give the movement range. Let the position and the estimated position of h_j be C and A, 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 in environment E_3. So CD is a fragment of the moving path in environment E_3. And in environment E_1, the positions of h_j and h_i are A and B. The moving path for h_j to approach h_i is AB.

Let $|DF| = R, |BE| = R, |AE| = L, |AC| = r, |EF| = r$ and A=(0,0), E=(L,0). Because the maximal value of $|CF|$ is achieved when C and F are on the circles centered at A and E respectively. Apply basic trigonometry, we get
$|CD| + R \leq |AB| + R + 2r$

If $P_3 : h_1 h_{3,2} \ldots h_{3,k-1}, h_k$ is a moving path in E_3, whose sum of moving length is s_3, then we have a moving path $P_4 : h_1 h_{4,2} \ldots h_{4,k-1}, h_k$ which has the length s_4 such that $s_4 - s_3 \leq (2k-2)r \leq (2n-2)r$. Because s_1 is the length of the shortest path among all moving paths from h_1 to h_k in E_1, we have $s_1 \leq s_4$. Thus we have $s_1 - s_3 \leq (2n-2)r$. □

By the above two lemmas, we have $s_2 - s_1 \leq (2k-3)r \leq (2n-3)r$ and $s_1 - s_3 \leq (2n-2)r$. In summary, $s_2 - s_3 \leq (4n-5)r$.