Message Relay in Disconnected Ad-hoc Networks

Qun Li and Daniela Rus
Department of Computer Science,
Dartmouth College,
Hanover, NH 03755
{liqun, 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 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.**

I. 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 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 trajectories of the hosts to transmit messages among hosts. We show how 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. 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, 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 ([4]). 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.

II. 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 [6]. 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.

Boukerche et al.[2], [1] proposed 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 will decrease. They also present a very nice analytical model based on Markov chains.

Another related area is PCS [5], [3] location management. 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 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.

III. 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. 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.

A. The Case of Multiple Messages

Suppose a set of hosts move according to prespecified 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 doesn't 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 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

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

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**

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 such that each host in the system can relay to the next host by modifying its moving 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 a system with low message rate, separately scheduling the route for each message is a reasonable approach.

B. *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 Figure 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.

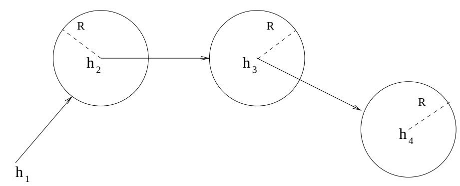

Fig. 1. In this figure, Node h_1 sends a message to h_4 by way of intermediate hosts h_2 and h_3. Disks corresponds to the transmission range of hosts and arrows show approach trajectories to relay messages.

The 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 (we describe this algorithm in Section III-B.1). The shortest trajectory (say 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 Figure 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.

Definition III.1: The motion of a host h_i is **predictable** if there is a known function $P_i(t)$ which de-

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 $h_1, h_2, \cdots, h_n$.

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 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

scribes 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, $h_0, h_2, ..., h_k$ (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 Figure 1, $h_1 h_2 h_3 h_4$ 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 $O(n^2 t)$ where t is the running time of the algorithm $OptimalTrajectory$.

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.

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

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

Theorem 1: The Optimal Relay Path algorithm (Algorithm 2) gives the optimal moving paths from host h_0 to all other hosts.

IV. 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 2: 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.

V. 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 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.

In this section we analyze the trade-offs between scope and update frequency in the MST protocol, by considering the error in a host's estimation about the location of another host. We consider 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.)

Suppose there are two hosts which have to communicate with each other, but they 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 nd 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 this time-varying behavior of a mobile host is Brownian motion with a drift process. The two dimensional Brownian motion with a drift process can be described by the distribution:

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

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: $r(t) = \sqrt{2D(t-t_0)\ln(\frac{1}{1-\gamma})}$. 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 ($l \geq R$), 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. 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 ($r \leq R$). 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:

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

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

Theorem 3: 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: $2 \cdot (\frac{P(l-R)}{2\lambda})^{\frac{1}{3}}$, $2 \cdot d^{\frac{1}{3}} \cdot \cos\frac{\theta}{3}$, 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 2 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.

Figure 2 (first) 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 2 (second) 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, 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, it takes $2 \cdot (\frac{P(l-R)}{2\lambda})^{\frac{1}{3}}$; when k increases, it takes $2 \cdot d^{\frac{1}{3}} \cdot \cos\frac{\theta}{3}$; when k is quite large it takes R.

The distance traveled by the hosts is determined by the length of a single trip and the number of trips. Figure 2 (second) shows that the bigger k is, the longer the optimal radius is. The reason is that for a large k (large

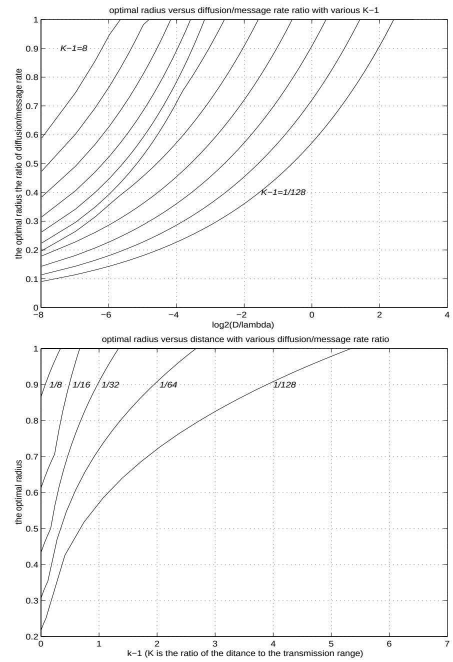

Fig. 2. This figure shows the optimal radius of the scope for the hosts. The left figure shows the dependency of this radius (represented by the y-axis) on the ratio $\frac{D}{\lambda}$. Each curve is drawn for different values of k, the ratio defined by distance between two hosts, divided by the transmission 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 $\frac{D}{\lambda}$

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 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.

VI. 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 the average transmission of a message.

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 the host in the waiting period are sent to the next host as a group if their next hosts are the same. Figures 3, 4, and 5 show the data we compiled from these experiments.

Figure 3 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 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.

Figure 4 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 the shorter travel path for a host, 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.

Figure 5 shows the influence of the various maximal speed values of the hosts on performance. It is obvious that a larger speed improves the performance.

VII. 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 method, which is not sufficient in some applications.

References

[1] A. Boukerche and S. K. Das. Message traffic control capabilities of the r-dsdv protocol in mobile ad hoc networks. In *Proc. ACM Modeling, Analysis and Simulation of Wireless and Mobile Systems, (MSWiM)*, pages 105–114, July 2001.

[2] A. Boukerche, S. K. Das, and A. Fabbri. Analysis of randomized congestion control with dsdv routing in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61:967–995, 2001.

[3] Azzedine Boukerche, Alessandro Fabbri, Oktay Yildiz, and Sajal K. Das. Performance analysis of a parallel pcs network simulation. In *HiPC*, pages 189–196, 1999.

[4] 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.

[5] 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.

[6] 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.

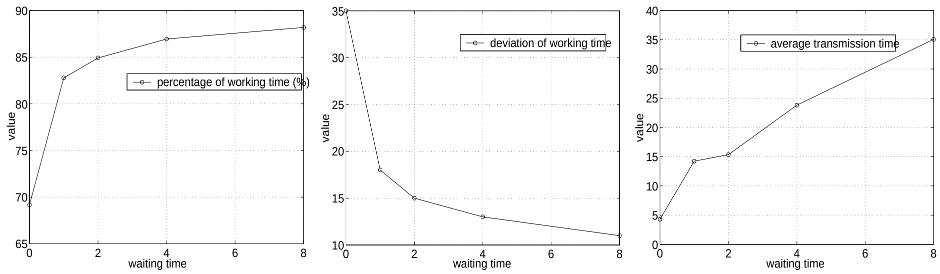

Fig. 3. The effect of varying the waiting time of hosts. The x-axis denotes the waiting time added to the hosts, while y-axis denotes the percentage of working time (first figure), the ratio of the standard deviation of the working times of the hosts and the average working time (second figure), and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

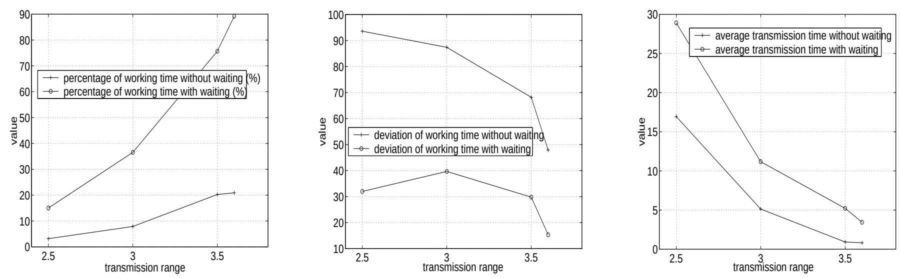

Fig. 4. The effect of varying the transmission range of hosts. The x-axis denotes the waiting time added to the hosts, while the y-axis denotes the percentage of working time (first figure), the ratio of the standard deviation of the working times of the hosts and the average working time (second figure), and the average transmission time of messages (third 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 is (2,0,20,2,0,0,20,20,20,5).

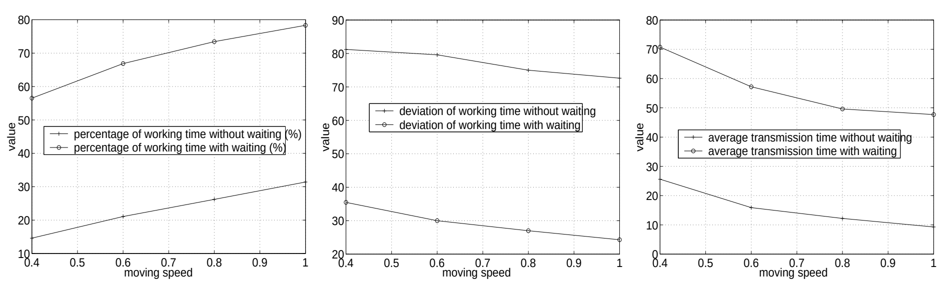

Fig. 5. The effect of varying the moving speed of hosts. The x-axis denotes the maximal moving speed of the hosts, while the y-axis denotes the percentage of working time (first figure), the ratio of the standard deviation of the working times of the hosts and the average working time (second figure), and the average transmission time of messages (third figure). It was simulated with 10 hosts, network space of 10*10, simulation time of 1000, message arrival rate of 2.0, transmission range of 3.0, and host moving speeds are 0.4, 0.6, 0.8, 1 separately. The waiting time vector is (2,0,15,2,0,0,1,20,15,0).