An Efficient Communication Strategy for Ad-hoc Mobile Networks*

I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis

[1] 1. Computer Technology Institute, Patras, Greece
{ichatz,nikole,spirakis}@cti.gr
[2] Computer Engineering and Informatics Department,
Patras University, Greece

Abstract. We investigate the problem of communication in an *ad-hoc* mobile network, that is, we assume the extreme case of a total absense of any fixed network infrastructure (for example a case of rapid deployment of a set of mobile hosts in an unknown terrain). We propose, in such a case, that a *small* subset of the deployed hosts (which we call the *support*) should be used for network operations. However, the vast majority of the hosts are moving *arbitrarily* according to application needs.
We then provide a simple, correct and efficient protocol for communication that avoids message flooding. Our protocol manages to establish communication between *any pair* of mobile hosts in *small, a-priori guaranteed expected time bounds* even in the *worst* case of *arbitrary* motions of the hosts that not in the support (provided that they do not deliberately try to avoid the support). These time bounds, interestingly, *do not depend*, on the number of mobile hosts that do not belong in the support. They depend only on the size of the area of motions. Our protocol can be implemented in very efficient ways by exploiting knowledge of the space of motions or by adding more power to the hosts of the support.
Our results exploit and further develop some fundamental properties of random walks in finite graphs.

1 Introduction

Ad-hoc Mobile Networks: An ad-hoc mobile network ([12,1]) is a collection of mobile hosts with wireless network interfaces forming a temporary network *without the aid of any established infrastructure or centralised administration.* In an ad-hoc network two hosts that want to communicate may not be within wireless transmission range of each other, but could communicate if other hosts between them in the ad-hoc network are willing to forward packets for them.

A *basic communication problem*, in such networks, is to send information from some *sender* user, S, to another designated *receiver* user, R. Remark that ad-hoc mobile networks are dynamic in nature, in the sense that local connections are temporary and may change as users move. The movement rate of each user might

* This work was partially supported by the EU projects IST FET-OPEN ALCOM-FT, IMPROVING RTN ARACNE and the Greek GSRT Project PENED99-ALKAD.

J. Welch (Ed.): DISC 2001, LNCS 2180, pp. 285–299, 2001.

vary, while certain hosts might even stop (even in "remote" areas) in order to execute location-oriented tasks (e.g. take measurements).

A protocol solving this important communication problem is *reliable* if it allows the sender to be notified about delivery of the information to the receiver.

The innovation and justification of our approach: One way to solve this problem is the protocol of notifying every user that the sender meets (and providing *all the information to it*) hoping that some of them will eventually meet the receiver.

> Is there a more efficient technique that will effectively solve the communication problem without flooding the network and exhausting the battery and computational power of the hosts?

The most common way to establish communication is to form paths of intermediate nodes that lie within one another's transmission range and can directly communicate with each other [13,15,16]. Indeed, this approach of exploiting pairwise communication is common in ad-hoc mobile networks that cover a relatively small space (i.e. with diameter which is small with respect to transmission range) or are dense (i.e. thousands of wireless nodes) where all locations are occupied by some hosts; broadcasting can be efficiently accomplished.

In wider area ad-hoc networks with less users, however, broadcasting is impractical: two distant peers will not be reached by any broadcast as users may not occupy all intermediate locations (i.e. the formation of a path is not feasible). Even if a valid path is established, single link "failures" happening when a small number of users that were part of the communication path move in a way such that they are no longer within transmission range of each other, will make this path invalid. Note also that the path established in this way may be very long, even in the case of connecting nearby hosts.

In contrast to all such methods, we try to avoid ideas based on paths finding and their maintenance. We envision networks with highly dynamic movement of the mobile users, where the idea of "maintenance" of a valid path is inconceivable (paths can become invalid immediately after they have been added to the directory tables). Our approach is to take advantage of the mobile hosts natural movement by exchanging information whenever mobile hosts meet incidentally. It is evident, however, that if the users are spread in remote areas and they do not move beyond these areas, there is no way for information to reach them, unless the protocol takes special care of such situations.

In the light of the above, we propose the idea of forcing only a small subset of the deployed hosts to move as per the needs of the protocol. Assuming the availability of such hosts, we use them to provide a simple, correct and efficient strategy for communication between any pair of hosts in such networks that avoid message flooding.

A scenario for rapid deployment of mobile hosts: A usual scenario that fits to the ad-hoc mobile model is the particular case of rapid deployment of mobile hosts, in an area where there is no underlying fixed infrastructure (either because it is impossible or very expensive to create such an infrastructure, or because it is not established yet, or it has become temporarily unavailable i.e. destroyed or down).

In such a case of rapid deployment of a number of mobile hosts, it is possible to have a small team of fast moving and versatile vehicles, to implement the support. These vehicles can be cars, jeeps, motorcycles or helicopters. We interestingly note that this small team of fast moving vehicles can also be a collection of independently controlled mobile modules, i.e. robots. This specific approach is inspired by the recent paper of J.Walter, J.Welch and N.Amato. In their paper "Distributed Reconfiguration of Metamorphic Robot Chains" ([17]) the authors study the problem of motion co-ordination in distributed systems consisting of such robots, which can connect, disconnect and move around. The paper deals with metamorphic systems where (as is also the case in our approach) all modules are identical. Note that the approach of having the support moving in a co-ordinated way, *i.e. as a chain of nodes*, has some similarities to [17].

Our results: We provide a particular protocol (and a specific support coordination subprotocol) which guarantees correct and efficient communication for any pair of users, in (expected) time *depending only on the size of the network area, independently of the motion of the hosts not in the support* and *independently of their number*. We achieve this by assuming that a *small* part of the deployed hosts, which we call the support, can move fast in a *coordinated* way, to *sweep* the motion space and act as an *intermediate pool* for receiving and delivering messages to the mobile users.

In a way similar to [17], these moving modules are identical in computing and communication (i.e. transmission) capability and run the same support management subprotocol to determine movement and communication of the hosts in the support. Furthermore, note that each module in the support needs only to know its current location (i.e. only local information is needed and not a global picture of the entire area). However, additional global information (such as knowledge of a spanning subgraph of the motion space) can improve the performance of our protocol.

Our protocol is simple, scalable, does not assume common sense of orientation, and does not need a lot of memory. It is resilient to single-host failures of the support. Furthermore, our protocol *avoids* the problem of flooding the network with messages.

The proof of our main theorem exploits the fundamental notion of strong stationary times of reversible Markov Chains. This notion allows us to consider general motion strategies of the users not in the support.

In [4] we performed extensive experiments (and some analysis) of a version of such a strategy but without the general framework and only for the *restricted case where all users (even those not in the support) perform independent and concurrent random walks*. A model for motion (without geometry details) for mobile networks was introduced by members of our team in [11]. Related material has appeared as a brief announcement in the Proceedings of the 20th Annual Symposium on Principles of Distributed Computing, [6]. For a survey of selected work in distributed communication and control issues in ad-hoc mobile networks, see [7].

Previous Work: In a recent paper [14], Q.Li and D.Rus present a model which has some similarities to ours. The authors give an interesting, yet different, protocol to send messages, which forces *all the mobile hosts to slightly deviate (for a short period of time) from their predefined, deterministic routes, in order to propagate the messages.* Their protocol is, thus, *compulsory* for any host and it works only for deterministic host routes. Moreover, their protocol considers the propagation of only one message (end to end) each time, in order to be correct. In contrast, our support scheme allows for simoultaneous processing of many communication pairs. In their setting [14] show optimality of message transmission times.

M.Adler and C.Scheideler [1] in a previous work, dealt only with *static* transmission graphs i.e. the situation where the positions of the mobile hosts and the environment do not change. In [1] the authors pointed out that static graphs provide a starting point for the dynamic case. In our work, we consider the *dynamic case* (i.e. mobile hosts move *arbitrarily*) and in this sense we extend their work. As far as performance is concerned, their work provides time bounds for communication that are proportional to the diameter of the graph defined by random uniform speading of the hosts, while our time bounds are linear to the area of motions, and *independent of the number of mobile hosts, or their spreading.*

We quantify our protocol's performance (in terms of communication *time*) and we show how to make it efficient and how to estimate the best size of the support.

2 The Model of the Space of Motions

Based on the work of [11,4] we abstract the environment where the stations move (in three-dimensional space with possible obstacles) by a *motion- graph* (i.e. we neglect the detailed geometric characteristics of the motion). In particular, we first assume that each mobile host has a transmission range represented by a sphere tr centred by itself. We approximate this sphere by a cube tc with volume $\mathcal{V}(tc)$ the maximum such that $\mathcal{V}(tc) < \mathcal{V}(tr)$. Given that the mobile hosts are moving in the space $\mathcal{S}$, $\mathcal{S}$ is divided into consecutive cubes of volume $\mathcal{V}(tc)$.

Definition 1. *The motion graph $G(V, E)$, ($|V| = n$, $|E| = m$), which corresponds to a quantization of $\mathcal{S}$ is constructed in the following way: a vertex $u \in G$ represents a cube of volume $\mathcal{V}(tc)$. An edge $(u, v) \in G$ if the corresponding cubes are adjacent.*

The number of vertices n, actually approximates the ratio between the volume of space $\mathcal{S}$, $\mathcal{V}(\mathcal{S})$, and the space occupied by the transmission range of a mobile host $\mathcal{V}(tr)$. Given the transmission range tr, n depends linearly on the volume of space $\mathcal{S}$ regardless of the choice of tc, and $n = O\big(\frac{V(\mathcal{S})}{V(tr)}\big)$. Let us call the ratio $\frac{V(\mathcal{S})}{V(tr)}$ by the term *relative motion space size* and denote it by ρ. Since the edges of G represent neighbouring polyhedra each node is connected with a constant number of neighbours, which yields that $m = \Theta(n)$. Let Δ be the maximum vertex degree of G.

3 A Protocol Framework for Ad-hoc Mobile Networks

We wish to look into ad-hoc networks where a small part of their hosts is used to serve network needs for communication. This is captured by the following:

Definition 2. *The class of ad-hoc mobile network protocols which enforce a (small) subset of the mobile hosts to move in a certain way is called the class of* semi-compulsory *protocols.*

Definition 3. *The subset of the mobile hosts of an ad-hoc mobile network whose motion is determined by a network protocol $\mathcal{P}$ is called the* support Σ *of $\mathcal{P}$. The part of $\mathcal{P}$ which indicates the way that members of Σ* move and communicate *is called the* support management subprotocol *of $\mathcal{P}$.*

Definition 4. *Consider a family of protocols, $\mathcal{F}$, for a mobile ad-hoc network, and let each $\mathcal{P}$ in $\mathcal{F}$ have the* same *support (and the same support management subprotocol). Then Σ is called the* support of the family $\mathcal{F}$.

In addition, we may wish that the way hosts in Σ move (maybe coordinated) and communicate is robust (i.e. can tolerate failures of hosts).

The types of failures of hosts that we consider here are permanent (i.e. stop) failures.

Definition 5. *A support management subprotocol, M_Σ, is k*-faults tolerant*, if it still allows the members of $\mathcal{F}$ (or $\mathcal{P}$) to execute correctly, under the presence of at most k permanent faults of hosts in Σ ($k \geq 1$).*

We assume, that the motions of the mobile users which are not members of Σ are arbitrary but *independent* of the motion of the support (i.e. we exclude the case where some of the users not in Σ are deliberately trying to avoid Σ). This is a pragmatic assumption usually followed by application protocols. We call it the *independence assumption.*

Definition 6. *A ad-hoc mobile network is* not hostile *if the hosts not in Σ obey the independence assumption.*

4 Our Proposed Strategy

4.1 The Scheme

Our proposed scheme, in simple terms, works as follows: The nodes of the support move fast enough in a coordinated way so that they sweep (in sufficiently short time) the entire motion graph. Their motion and communication is accomplished in a distributed way via a *support management subprotocol* M_{Σ}. When some node of the support is within communication range of a sender, an underlying *sensor subprotocol* M'_{Σ} notifies the sender that it may send its message(s).

The messages are then stored "somewhere within the support structure". For simplicity we may assume that they are copied and stored in every node of the support. This is not the most efficient storage scheme and can be refined in various ways. When a receiver comes within communication range of a node of the support, the receiver is notified that a message is "waiting" for him and the message is then forwarded to the receiver. For simplicity, we will also assume that message exchange between nodes within communication distance of each other takes negligible time. Note that this general scheme allows for easy implementation of many-to-one communication and also multicasting. In a way, the support Σ plays the role of a (moving) skeleton subnetwork (of a "fixed" structure, guaranteed by the motion subprotocol M_{Σ}), through which all communication is routed. From the above description, the size, k, and the shape of the support may affect performance.

Our scheme follows the general design principle of mobile networks (with a fixed subnetwork however) called the "two-tier" principle ([12]) which says that any protocol should try to move communication and computation to the fixed part of the network. Our idea of the support Σ is a simulation of such a (skeleton) network by moving hosts, however.

Note that the proposed scheme does not require the propagation of messages through hosts that are not part of Σ, thus its security relies on the support's security and is not compromised by the participation in message communication of other mobile users. For a discussion of intrusion detection mechanisms for ad-hoc mobile networks see [18].

4.2 The Implementation Proposed for Σ, M_{Σ}

There is a set-up phase of the ad-hoc network, where a predefined set, k, of hosts, become the nodes of the support. The members of the mobile support perform a leader election by running a randomized symmetry breaking protocol in anonymous networks ([11]). This imposes only an initial communication cost. The elected leader, denoted by MS_0, is used to co-ordinate the support topology and movement. Additionally, the leader assigns local names to the rest of the support members (MS_1, MS_2, ... , MS_{k-1}). The movement of Σ is then defined as follows:

> Initially, MS_i, $\forall i \in \{0, 1, \ldots, k-1\}$, start from the same area-node of the motion graph. The direction of movement of the leader MS_0 is

given by a memoryless operation that chooses *randomly the direction* of the next move. Before leaving the current area-node, MS_0 sends a message to MS_1 that states the new direction of movement. MS_1 will change its direction as per instructions of MS_0 and will propagate the message to MS_2. In analogy, MS_i will follow the orders of MS_{i-1} after transmitting the new directions to MS_{i+1}. Movement orders received by MS_i are positioned in a queue Q_i for sequential processing. The very first move of MS_i, $\forall i \in \{1, 2, \ldots, k-1\}$ is delayed by δ period of time.

We assume that the mobile support hosts move with a common speed. Note that the above described motion subprotocol M''_Σ enforces the support to move as a "snake", with the head (the elected leader MS_0) *doing a random walk on the motion graph* G and each of the other nodes MS_i executing the simple protocol "move where MS_{i-1} was before". Therefore our protocol does not require common sense of orientation.

The purpose of the random walk of the head is to ensure a *cover* (within some finite time) of the whole motion graph, without memory (other than local) of topology details. Note that this memoryless motion also ensures fairness.

A modification of M_Σ is that the head does a random walk on *a spanning subgraph* of G (eg. a spanning tree). This modified M_Σ (call it T_Σ) is more efficient in our setting since "edges" of G just represent adjacent locations and "nodes" are really possible host places.

4.3 Alternative Implementations - Extensions

One can think also of other ways to implement the support management subprotocol M_Σ:

- The *runners* implementation of M_Σ allows each member of Σ to move via an *independent* random walk (on the same spanning subgraph of G). When runners meet, they exchange information given to them by hosts. This management subprotocol provides improved reliability in the sense that it is resilient to t faults, where $t < k$. However, note that messages may have to be re-transmitted in the case that only one copy of them exists when the faults occur.

The key observation justifying this approach (and maybe its superiority, with respect to performance, compared to the "snake" approach) is that each runner will meet each other in parallel, thus accelerating the spread of information. In [8] we experimentally showed that the "runners" protocol outperforms the "snake" protocol.

- In *hierarchical* motion graphs [5] we can divide Σ into a subset Σ' moving only in the upper level of the hierarchy and the hosts of $\Sigma - \Sigma'$ which can be split in "snakes", each randomly walking inside the lower levels of the hierarchy. The lower level of the hierarchy may model dense ad-hoc subnetworks of mobile users that are unstructured and where there is no fixed infrastructure. To implement communication in such a case, a possible solution would be to install a very fast (yet limited) backbone interconnecting such highly populated mobile user areas, while using the support approach in the lower levels. This fast backbone provides a limited number of access ports within these dense areas of mobile users.

In such hierarchical cases communication between users in different dense areas takes place in the following way: The support first gets from the sender node the messages upon meeting him and conveys these messages to the backbone system when meeting the corresponding access port. Then by exploiting the very fast communication over the backbone, Σ_1 forwards the messages to some access port in the receiver area, from which subsequently the messages are picked by the local support (Σ_2) and delivered to the receiver host.

We note that this hierarchical approach for a management subprotocol is inherently *modular*.

4.4 Protocol Correctness Properties

In the sequel we investigate non-hostile ad-hoc mobile networks. We assume that each mobile host has sufficient power supplies (or on-line power feedings) to support communication for long times. Moreover, we assume (to simplify the technical analysis) common speed and fixed transmission range for the hosts not in the support.

In the sequel, we assume that the head of Σ does a *continuous time random walk* on $G(V, E)$, without loss of generality (we can discretize). We define the random walk of a mobile user on G that induces a continuous time Markov chain M_G as follows: The states of M_G are the vertices of G. Let s_t denote the state of M_G at time t. Given that $s_t = u$, $u \in V$, the probability that $s_{t+dt} = v$, $v \in V$, is $p(u, v) \cdot dt$ where

$$p(u,v) = \begin{cases} \frac{1}{d(u)} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

and $d(u)$ is the degree of vertex u.

Definition 7. *$P_i(E)$ is the probability that the walk satisfies an event E given it started at vertex i.*

Definition 8. *For a vertex j, let T_j be the first hitting time of the walk onto that vertex and let $E_i T_j$ be its expected value, given that the walk started at vertex i of G.*

Definition 9. *For the walk of Σ's head, let $\pi()$ be the stationary distribution of its position after a sufficiently long time.*

We know (see [2]) that for every vertex σ, $\pi(\sigma) = \frac{d(\sigma)}{2m}$ where $d(\sigma)$ is the degree of σ in G and $m = |E|$.

Definition 10. *Let $p_{j,k}$ be the transition probability of the walk of Σ's head from vertex j to vertex k. Let $p_{j,k}(t)$ be the probability that the walk started at j will be at $k \in V$ in time t.*

Theorem 1. *The support Σ and the management subprotocol M_Σ guarantee reliable communication establishment between any sender-receiver (S, R) pair in finite time, whose expected value is bounded only by a function of the relative motion space size ρ and does not depend on the number of hosts, and is also independent of how S, R move.*

Proof. Any sender S or receiver R is allowed an *arbitrary* strategy of motion but it does not deliberately try to avoid the support Σ. So, it either executes a deterministic motion (which either stops at a node, or repeats forever) or follows a random strategy *independent* of the random walk of the support's head.

For the proof purposes, it is enough to show that the head of Σ will meet S and R infinitely often, with probability 1 (in fact our argument is a consequence of the Borel-Cantelli Lemmas for infinite sequences of trials). We will furthermore show that the first meeting time M (with S or R) has an expected value (where expectation is taken over the walk of Σ and any strategy of S (or R) and any starting position of S (or R) and Σ) which is bounded by a function of the size of the motion graph G only. This then shows the Theorem since it shows that S (and R) meet with the head of Σ infinitely often, each time within a bounded expected duration.

So, let EM be the expected time of the (first) meeting and $m^* = supEM$, where the supremum is taken over all starting positions of both Σ and S (or R) and all strategies of S (one can repeat the argument with R).

We will now assume w.l.o.g. (see [2]) that the head of Σ's walk is a continuous-time random walk on G. The states of the walk of Σ's head are just the vertices of G and they are finite.

Definition 11. *Let $X(t)$ be the position of the walk at time t*

We proceed to show that we can construct for the walk of Σ's head a *strong stationary time sequence* V_i such that for all $\sigma \in V$ and for all times t

$$P_i(X(V_i) = \sigma \ | \ V_i = t) \ = \ \pi(\sigma)$$

Notice that at times V_i, S (or R) will necessarily be at some vertex σ of V, either still moving or stopped. Let u be a time such that for X,

$$p_{j,k}(u) \geq \left(1 \ - \ \frac{1}{e}\right)\pi(k)$$

for all j, k. Such a u always exists because $p_{j,k}(t)$ converges to $\pi(k)$ from basic Markov Chain Theory. Note that u depends only on the structure of the walk's graph, G. In fact, if one defines separation from stationarity to be

$$s(t) = max_j s_j(t)$$

where

$$s_j(t) = sup\{s : p_{ij}(t) \ \geq \ (1 \ - \ s)\pi_j \}$$

then

$$\tau_1^{(1)} = min\{t : s(t) \leq e^{-1}\}$$

is called the *separation threshold time*. For general graphs G of n vertices this quantity is known to be $\mathcal{O}(n^3)$ ([3]).

Now consider a sequence of stopping times $U_i \in \{u, 2u, 3u, \dots\}$ such that

$$P_i(X(U_i) = \sigma \mid U_i = u) = \left(1 - \frac{1}{e}\right)\pi(\sigma) \tag{1}$$

for any $\sigma \in V$. By induction on $\lambda \geq 1$ then

$$P_i\big(X(U_i) = \sigma \mid U_i = \lambda u\big) = e^{-(\lambda-1)}\left(1 - \frac{1}{e}\right)\pi(\sigma)$$

This is because of the following: First remark that for $\lambda = 1$ we get the definition of U_i. Assume that the relation holds for $(\lambda - 1)$ i.e.

$$P_i\big(X(U_i) = \sigma \mid U_i = (\lambda-1)u\big) = e^{-(\lambda-2)}\left(1 - \frac{1}{e}\right)\pi(\sigma)$$

for any $\sigma \in V$. Then $\forall \sigma \in V$

$$\begin{aligned} P_i\big(X(U_i) = \sigma \mid U_i = \lambda u\big) &= \sum_{\alpha \in V} P_i\big(X(U_i) = \alpha \mid U_i = (\lambda-1)u\big) \cdot P_{\alpha,\sigma}(u) \\ &= e^{-(\lambda-2)}\left(1 - \frac{1}{e}\right)\sum_{\alpha \in V}\pi(\alpha)\frac{1}{e}\pi(\sigma) \quad \text{from (1)} \\ &= e^{-(\lambda-1)}\left(1 - \frac{1}{e}\right)\pi(\sigma) \end{aligned}$$

which ends the induction step. Then, for all σ

$$P_i\big(X(U_i) = \sigma\big) = \pi(\sigma) \tag{2}$$

and

$$E_i U_i = u\left(1 - \frac{1}{e}\right)^{-1}$$

Now let $c = u\frac{e}{e-1}$. So, we have constructed (by (2)) a *strong stationary time sequence* U_i with $EU_i = c$. Consider the sequence $0 = U_0 < U_1 < U_2 < \dots$ such that for $i \geq 0$

$$E\big(U_{i+1} - U_i \mid U_j,\ j \leq i\big) \leq c$$

But, from our construction, the positions $X(U_i)$ are *independent* (of the distribution $\pi()$), and, in particular, $X(U_i)$ *are independent* of U_i. Therefore, regardless of the strategy of S (or R) and because of the independence assumption, the support's head has chance at least $\min_\sigma \pi(\sigma)$ to meet S (or R) at time U_i, independently as i varies. So, the meeting time M satisfies $M \leq U_T$ where T is a stopping time with mean

$$ET \leq \frac{1}{min_\sigma \pi(\sigma)}$$

Note that the idea of a stopping time T such that $X(T)$ has distribution π and is independent of the starting position is central to the standard modern Theory of Harris - recurrent Markov Chains (see e.g. [10]).

¿¿From Wald's inequality ([2]) then $EU_T \ \leq \ c \cdot ET$, thus

$$m^* \ \leq \ c \ \frac{1}{\min_\sigma \pi(\sigma)}$$

Note that since G is produced as a subgraph of a regular graph of fixed degree Δ we have

$$\frac{1}{2m} \ \leq \ \pi(\sigma) \ \leq \ \frac{1}{n}$$

for all σ (n=|V|, m=|E|), thus $ET \ \leq \ 2m$, hence

$$m^* \leq 2mc \ = \ \frac{e}{e-1} 2mu$$

Since m, u only depend on G, this proves the Theorem.

□

Corollary 1. *If Σ's head walks randomly in a* regular *spanning subgraph of G, then $m^* \leq 2cn$.*

Now, we examine the robustness of the motion management subprotocol under single stop-faults.

Theorem 2. *The support management subprotocol M_Σ is 1-fault tolerant.*

Proof. If a single host of Σ fails, then the following host becomes the head of the rest of the "snake". We thus have two, independent, random walks in G (of the two "snakes") which, however, will meet in expected time at most m^* (as in Theorem 1) and re-organize as a single snake via a very simple re-organization protocol which is the following:

When the head of the second snake Σ_2 meets a host h of the first snake Σ_1 then the head of Σ_2 follows the host which is "in front" of h in Σ_1, and all the part of Σ_1 after and including h waits, to follow Σ_2's tail.

□

Note that in the case when more than one faults occur, the procedure for merging "snakes" described above may lead to deadlock, as figure 1 graphically depicts.

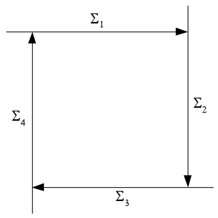

Fig. 1. Deadlock situation arising when four "snakes" are about to merge.

4.5 Protocol Time Efficiency Properties

Crude bounds. Clearly, one intuitively expects that if $k = |\Sigma|$ then the higher k is (with respect to n), the best the performance of Σ gets.

By working as in the proof of Theorem 1, we can create a sequence of strong stationary times U_i such that $X(U_i) \in F$ where $F = \{\sigma : \sigma$ is a position of a host in the support$\}$. Then $\pi(\sigma)$ is replaced by $\pi(F)$ which is just $\pi(F) = \sum \pi(\sigma)$ over all $\sigma \in F$. So now m^* is bounded as follows:

$$m^* \leq c \frac{1}{\min_{\sigma \in J} \left(\sum \pi(\sigma) \right)}$$

where J is any induced subgraph of the graph of the walk of Σ's head such that J is the neighbourhood of a vertex σ of radius (maximum distance simple path) at most k. The quantity

$$\min_J \left(\sum_{\sigma \in J} \pi(\sigma) \right)$$

is then at least $\frac{k}{2m}$ and, hence, $m^* \leq c \frac{2m}{k}$.

Since the communication establishment time, T_c, between S, R is bounded above by X+Y+Z, where X is the time for S to meet Σ, Y is the time for R to meet Σ (after X) and Z is the message propagation time in Σ, we have for all S, R

$$E(T_c) \leq \frac{2mc}{k} + \Theta(k) + \frac{2mc}{k}$$

(since $Z = \Theta(k)$). The upper bound achieves a minimum when $k = \sqrt{2mc}$.

Lemma 1. *For the walk of Σ's head on the entire motion graph G, the communication establishment time's expected time is bounded above by $\Theta(\sqrt{mc})$ when the (optimal) support size $|\Sigma|$ is $\sqrt{2mc}$ and c is $\frac{e}{e-1}u$, u being the "separation threshold time" of the random walk on G.*

Tighter bounds - improved protocol. To make our protocol more efficient, we now force the head of Σ to perform a random walk on a *regular spanning graph* of G. Let $G_R(V, E')$ be such a subgraph. Our improved protocol versions assume that (a) such a subgraph exists in G and (b) is given in the beginning to all the stations of the support. By studying, in a way similar to Theorem 1 and [2], the first meeting times and the separation from stationarity of the random walk on the regular spanning graph, we get the following theorem (for the proof see [9,7]) :

Theorem 3. *By having Σ's head to move on a regular spanning subgraph of G, there is an absolute constant $\gamma > 0$ such that the expected meeting time of S (or R) and Σ is bounded above by $\gamma \frac{n^2}{k}$.*

Remark again that the total expected communication establishment time is bounded above by $2\gamma\frac{n^2}{k} + \Theta(k)$ and by choosing $k = \sqrt{2\gamma n^2}$ we can get a best bound of $\Theta(n)$ for a support size of $\Theta(n)$.

Corollary 2. *By forcing the support's head to move on a regular spanning subgraph of the motion graph, our protocol guarantees a total expected communication time of $\Theta(\rho)$, where ρ is the relative motion space size, and this time is independent of the total number of mobile hosts, and their movement.*

Note also that our analysis assumed that the head of Σ moves according to a continuous time random walk of total rate 1 (rate of exit out of a node of G). If we select the support's hosts to be ψ *times faster* than the rest of the hosts, all the estimated times, except of the inter-support time, will be divided by ψ. Thus

Corollary 3. *Our modified protocol where the support is ψ times faster than the rest of the mobile hosts guarantees an expected total communication time which can be made to be as small as $\Theta\left(\gamma \frac{\rho}{\sqrt{\psi}}\right)$ where γ is an absolute constant.*

5 A Lower Bound

Lemma 2.

$$m^* \geq \max_{i,j} E_i T_j$$

Proof. Consider the case where S (or R) just stands still on some vertex j and Σ's head starts at i. □

Corollary 4. *When Σ starts at positions according to the stationary distribution π of its head's walk then, $\forall j$,*

$$m^* \geq \max_{j} E_\pi T_j$$

□

From a Lemma of ([2], ch. 4, pp. 21), we know that for all i

$$E_\pi T_i \ \geq \ \frac{(1 \ - \ \pi_i)^2}{q_i \pi_i}$$

where $q_i = d_i$ is the degree of i in G i.e.,

$$E_\pi T_i \ \geq \ \min_i \frac{\left(1 \ - \ \frac{d_i}{2m}\right)^2}{d_i \frac{d_i}{2m}} \geq \ \min_i \frac{1}{2m} \frac{(2m \ - \ d_i)^2}{d_i^2}$$

For regular spanning subgraphs of G of degree Δ we have $m = \frac{\Delta n}{2}$, where $d_i = \Delta$ for all i. Thus,

Theorem 4. *When Σ's head moves on a regular spanning subgraph of G, of m edges, we have that the expected meeting time of S (or R) and Σ cannot be less than* $\frac{(n \ - \ 1)^2}{2m}$.

Corollary 5. *Since $m = \Theta(n)$ we get a $\Theta(n)$* lower bound *for the expected communication time. In that sense, our protocol's expected communication time is* optimal *when the support size is $\Theta(n)$.*

6 Extensions of Our Work

First of all we notice that our work does not assume any particular motion of hosts not in Σ (other than that we are in non-hostile networks). We pose as an open problem the notion of "capture" of S (or R) in hostile networks. We also remark that any assumption on motions of hosts $s \notin \Sigma$ will lead to much better upper bounds on the communication time. We plan to investigate the case of varying transmission ranges. We also pose as an open problem the proof of correctness and the efficiency analysis of the proposed alternative implementations, and especially the analytic comparison of the "snake" and the "runners" approach performance. Finally, it is interesting to comparatively study the performance of our approach versus other routing protocols (such as TORA, AODV, LAR) through experiments.

References

1. M. Adler and C. Scheideler: Efficient Communication Strategies for Ad-Hoc Wireless Networks. In Proc. 10th Annual Symposium on Parallel Algorithms and Architectures (SPAA'98)(1998).
2. D. Aldous and J. Fill: *Reversible Markov Chains and Random Walks on Graphs.* Unpublished manuscript. http://stat-www.berkeley.edu/users/aldous/book.html (1999).
3. G. Brightwell, P. Winkler: Maximum Hitting Times for Random Walks on Graphs. Journal of Random Structures and Algorithms, 1:263-276, 1990 (1990).

4. I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: Analysis and Experimental Evaluation of an Innovative and Efficient Routing Approach for Ad-hoc Mobile Networks. In Proc. 4th Annual Workshop on Algorithmic Engineering (WAE'00)(2000).
5. I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: An Efficient Routing Protocol for Hierarchical Ad-Hoc Mobile Networks In Proc. 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 15th Annual International Parallel & Distributed Processing Symposium (IPDPS'01)(2001).
6. I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: An Efficient Communication Strategy for Ad-hoc Mobile Networks. As Brief Announcement in Proc. 20th Annual Symposium on Principles of Distributed Computing, (PODC'01)(2001).
7. I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: On the Average and Worst-case Efficiency of Some New Distributed Communication and Control Algorithms for Ad-hoc Mobile Networks. Invited Paper in Proc. 1st International Workshop on Principles of Mobile Computing, (POMC'01)(2001).
8. I. Chatzigiannakis, S. Nikoletseas, N. Paspalis, P. Spirakis and C. Zaroliagis: An Experimental Study of Basic Communication for Ad-hoc Mobile Networks. In Proc. 5th Annual Workshop on Algorithmic Engineering (WAE'01)(2001).
9. I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: An Efficient Communication Strategy for Ad-hoc Mobile Networks (full paper). CTI Technical Report, July 2001, 2001. http://faethon.cti.gr/adhoc/disc01.html
10. R. Durret: *Probability: Theory and Examples*. Wadsworth. (1991).
11. K. P. Hatzis, G. P. Pentaris, P. G. Spirakis, V. T. Tampakas and R. B. Tan: Fundamental Control Algorithms in Mobile Networks. In Proc. 11th Annual Symposium on Parallel Algorithms and Architectures (SPAA'99) (1999)
12. T. Imielinski and H. F. Korth: *Mobile Computing*. Kluwer Academic Publishers. (1996).
13. Y. Ko and N. H. Vaidya: Location-Aided Routing (LAR) in Mobile Ad-hoc Networks. In Proc. 4th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'98) (1998).
14. Q. Li and D. Rus: Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks. In Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'00) (2000).
15. V. D. Park and M. S. Corson: Temporally-ordered routing algorithms (TORA) version 1 functional specification. IETF, Internet Draft, draft-ietf-manet-tora-spec-02.txt, Oct. 1999. (1999).
16. C. E. Perkins and E. M. Royer: Ad-hoc on demand distance vector (AODV) routing. IETF, Internet Draft, draft-ietf-manet-aodv-04.txt (IETF'99). (1999).
17. J. E. Walter, J.L. Welch and N. M. Amato: Distributed Reconfiguration of Metamorphic Robot Chains In Proc. 19th Annual Symposium on Principles of Distributed Computing (PODC'00) (2000).
18. Y. Zhang and W. Lee: Intrusion Detection in Wireless Ad-hoc Networks. In Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'00) (2000).