An Efficient Routing Protocol for Hierarchical Ad-hoc Mobile Networks *

I.Chatzigiannakis, S.Nikoletseas and P.Spirakis

Computer Technology Institute,
Patras, Greece

Computer Engineering and Informatics Department,
Patras University,
Greece

E-mail: {ichatz,nikole,spirakis}@cti.gr

Abstract

We introduce a new model *of ad-hoc mobile networks, which we call* hierarchical*, that are comprised of dense subnetworks of mobile users (corresponding to highly populated geographical areas, such as* cities*), interconnected across access ports by sparse but frequently used connections (such as* highways*).*

For such networks, we present an efficient routing protocol *which extends the idea (introduced in [4]) of exploiting the co-ordinated motion of a* small part *of an ad-hoc mobile network (the "support") to achieve very fast communication between any two mobile users of the network. The basic idea of the new protocol presented here is, instead of using a unique (large) support for the whole network, to employ* a hierarchy of (small) supports *(one for each city) and also* take advantage of the regular traffic *of mobile users across the interconnection highways to communicate between cities.*

We combine here theoretical analysis (average case estimations based on random walk properties) and experimental implementations (carried out using the LEDA platform) to claim and validate results showing that such a hierarchical routing approach is, for this class of ad-hoc mobile networks, significantly more efficient *than a simple extension of the basic "support" idea presented in [4].*

1 Introduction, State of the Art and Our Results

Mobile computing has been introduced (mainly as a result of major technological developments) in the past few years forming a new computing environment. Because of the fact that mobile computing is constrained by poor resources, highly dynamic variable connectivity and restricted energy sources, the design of stable and efficient mobile information systems has been greatly complicated. Until now, two basic system models have been proposed for mobile computing. The "fixed backbone" mobile system model has been around the past decade and has evolved to a fairly stable system that can exploit a variety of information in order to enhance already existing services and yet provide new ones. On the other hand, the "ad hoc" system model assumes that mobile hosts can form networks without the participation of any fixed infrastructure.

An ad hoc mobile network ([9]) 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 are also participating in the ad hoc network and are willing to forward packets for them.

Suppose that mobile hosts equipped with wireless transmitters and receivers are moving in a geographical area forming an ad hoc network. Suppose further that these hosts want to execute a simple distributed protocol such as leader election. One way to perform this is to utilise an underlying communication protocol (see [2]), which delivers (if possi-

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

ble) a message from one mobile host to another, regardless of their position. This scheme, in the case of high mobility of the hosts, could lead to a situation where most of the computational and battery power of the hosts is consumed for the communication protocol.

> Is there a more efficient technique (other than notifying every station that the sender meets, in the hope that some of them will then eventually meet the receiver) that will effectively solve the routing problem without flooding the network and exhausting the battery and computational power of the hosts?

Routing between two hosts in a mobile ad hoc network has been a crucial problem and several approaches have been developed. Its most important performance characteristic is the amount of (routing related) traffic generated regarding the mobility rating of the hosts. Conventional routing protocols are insufficient for ad hoc networks since the routing information generated after each topology change may waste a large portion of the wireless bandwidth. In [3] J. Broch, D. Johnson and A. Maltz propose the Dynamic Source Routing (DSR) protocol, which uses on-demand route discovery. There exist many variations of the DSR protocol that try to optimise the route discovery overhead. [15] presents the AODV (Ad Hoc On Demand Distance Vector routing) protocol that also uses a demand-driven route establishment procedure. More recently TORA (Temporally-Ordered Routing Algorithm, [14]) is designed to minimise reaction to topological changes by localising routing-related messages to a small set of nodes near the change. [5] and [6] attempt to combine proactive and reactive approaches in the Zone Routing Protocol (ZRP), by initiating route discovery phase on-demand, but limit the scope of the proactive procedure only to the initiator's local neighbourhood. [11] propose the Location-Aided Routing (LAR) protocol that uses a global positioning system to provide location information that is used to improve the performance of the protocol by limiting the search for a new route in a smaller "request zone".

In [4] we have presented an innovative, efficient routing protocol based on the idea of using the co-ordinated motion of *a small, "snake-like"* part of an ad-hoc mobile network (the so-called "support") sweeping randomly the network and acting as an intermediate message store-and-forward fixed infrastructure. This protocol, which we briefly describe in the next section, achieves very fast communication times between any two mobile users of the network.

In this work, we introduce a *new model* of ad-hoc mobile networks, which we call *hierarchical*, that are comprised of dense subnetworks of mobile users (corresponding to highly populated geographical areas, such as *cities*), interconnected across access ports by sparse but frequently used connections (such as *highways*).

For such networks, we present an *efficient routing protocol* which extends the idea (introduced in [4]) of exploiting the co-ordinated motion of a small part of an ad-hoc mobile network (the "support") to achieve very fast communication between any two mobile users of the network. The basic idea of the new protocol presented here is, instead of using a unique (large) support for the whole network, to employ *a hierarchy of (small) supports* (one for each city) and also *take advantage of the regular traffic* of mobile users across the interconnection highway to communicate between cities.

We combine here theoretical analysis (average case estimations based on random walk properties) and experimental implementations (carried out using the LEDA platform) to claim and validate results showing that such a hierarchical routing approach is, for this class of ad-hoc mobile networks, *significantly more efficient* than a simple extension of the basic "support" idea presented in [4].

2 Our Previous Work

In [4] we have proposed, theoretically analyzed and experimentally validated an innovative routing protocol that can be efficiently applied to weaker models of ad-hoc networks as no location information is needed neither for the support or for any other host of the network. Additionally, our protocol does not use conventional methods for path finding; the highly dynamic movement of the mobile hosts can make the location of a valid path inconceivable - paths can become invalid immediately after they have been added to the directory tables.

Based on the work of [7, 4] we use a graph theoretic concept where the movement of the mobile users in the three-dimensional space S is mapped to a motion graph G(V,E), $|V|$=n, $|E|$=e.

In [4] we provide a particular implementation of efficient node-to-node communication in a mobile ad-hoc network by introducing the idea of a (mobile) small-sized *support* subnetwork i.e. a subset of nodes that move in a coordinated way and act as an intermediate storage of messages. Our proposed support is a "snake-like" sequence of stations that always remains pair-wise adjacent and move in a way determined by the snake's head. The head moves by executing a random walk on the motion graph.

Definition 1 *The* support, Σ, *of an ad-hoc mobile network is a subset of the mobile hosts, moving in a* co-ordinated way *and always remaining pairwise adjacent, as indicated by the* support motion subprotocol*.*

Thus, this protocol can be called "semi-compulsory" in the sense that it forces a subset of the mobile hosts to move in a co-ordinated way.

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

The scheme proposed in [4], in simple terms, works as follows: The nodes of the support move in a coordinated way so that they sweep (given some time) the entire motion graph. Their motion is accomplished in a distributed way via a *support motion subprotocol* P_1. When some node of the support is within communication range of a sender, an underlying *sensor subprotocol* P_2 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 (i.e. the messages are short packets). 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 P_1), through which all communication is routed. From the above description, the size, k, and the shape of the support may affect performance.

A more detailed implementation description of the protocol [4] follows:

At the set-up phase of the ad-hoc network, a predefined number, k, of hosts, become the nodes of the support. The members of the mobile support perform a leader election, which is run once and 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, $\forall i \in \{0, 1, ..., k-1\}$, 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, $\forall i \in \{1,2,...,k-1\}$ 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 P_1 enforces the support to move as a "snake", with the head (the elected leader MS_0) doing a random walk on the motion graph and each of the other nodes MS_i executing the simple protocol "move where MS_{i-1} was before". This can be easily implemented because MS_i will move following the edge from which it received the message from MS_{i-1} and therefore our protocol does not require common sense of orientation.

In the experiments done as part of the [4] paper, we noticed that the communication times are slightly improved when the head of the snake excludes from its random choice its current position. We also noticed that only negligible gains come by having MS_0 to "remember and avoid" even many of previous snake positions.

In particular we analytically proved and experimentally validated in [4] the following basic result for the expected communication time between any two mobile users of the network, where G is the motion graph, $\lambda_2(G)$ is its second eigenvalue, n is the number of vertices in G and k is the size of the support:

Theorem 1 *([4]) The expected communication time of our routing protocol is bounded above by the formula*

$$E(T_{total}) \;\leq\; \frac{2}{\lambda_2(G)}\,\Theta\left(\frac{n}{k}\right) \;+\; \Theta(k)$$

3 The Model: Hierarchical Ad-hoc Mobile Networks

In this work, we first introduce a new, hierarchical model of ad-hoc mobile networks, strongly motivated by real life situations.

Such hierarchical ad-hoc mobile networks are basically comprised of sparse but frequently traversed interconnections of dense subnetworks of mobile users. These dense subnetworks may appear in cases of high concentration of mobile users, such as highly populated geographical areas (cities). We abstract such dense subnetworks of mobile users by *city graphs*.

In particular, we abstract the environment where the users move (in three-dimensional space with possible obstacles) by a *motion-graph* (i.e. we neglect the detailed geometric characteristics of the motion. We expect that future research will incorporate geometry constraints into the subject). In particular, we first assume (as in [7]) that each mobile host has a transmission range represented by a sphere tr centred by itself. This means that any other host inside tr can receive any message broadcasted by this host. We approximate this sphere by a cube tc with volume $\mathcal{V}(tc)$, where $\mathcal{V}(tc) < \mathcal{V}(tr)$. The size of tc can be chosen in such a way that its volume $\mathcal{V}(tc)$ is the maximum that preserves $\mathcal{V}(tc) < \mathcal{V}(tr)$, and if a mobile host inside tc broadcasts a message, this message is received by any other host in tc. Given that the mobile hosts are moving in the space S, S is divided into consecutive cubes of volume $\mathcal{V}(tc)$.

Such dense ad-hoc networks of mobile users are usually interconnected across specific access points or ports in a sparse (and thus efficient) way by certain connections (such as highways). Such interconnection highways, although sparse, exhibit a more or less high and regular traffic (as an example, because of many people and vehicles frequently moving from one city to another).

We thus distinguish two different categories of mobile users in such hierarchical ad-hoc mobile networks:

Definition 3 *A* highway mobile user *is a mobile user doing only traversal of the intercity sparse subnetwork, i.e. moves only on the interconnection highways passing frequently from the access ports.*

Definition 4 *A* city mobile user *is a mobile user which randomly moves in a city's area, i.e. performs a random walk on the city's motion graph.*

Because of the regular traffic of highway mobile users in the interconnection highways, we assume that in each specific moment in time the probability of a highway mobile user being at a city's access port is p.

Definition 5 *Let p be the probability that at any given time instance there is a highway mobile user at a city's access port.*

Note 1 *This probability p can also model the intercity exchange of information via other means (e.g. fixed infrastructure, satellites etc.).*

We also remark that, in practice, this probability may differ between the various access ports and also vary with time. So we take, for analysis reasons, p to be a lower bound on these probabilities.

Figure 1 shows a graphical representation of hierarchical ad-hoc mobile networks.

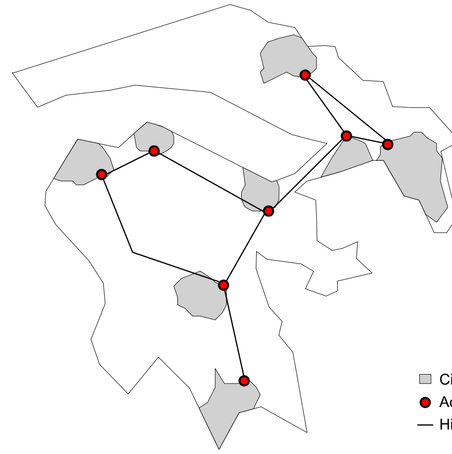

Figure 1. A graphical representation of a hierarchical ad-hoc mobile network made up from eight dense subnetworks (cities).

4 The Hierarchical Support Routing Protocol (HSRP)

In this work we exploit the hierarchical structure of such networks (by employing one support in each city) and also take advantage of this regular (not random) movement of mobile users in the interconnection highways to achieve significantly faster communication than by simply extending the protocol in [4] (i.e. having a unique, large support for the whole network).

More specifically, we propose the following hierarchical extension of the basic idea of a support introduced in [4]. In each city (dense ad-hoc subnetwork of mobile users) we use a support of size implied by the analysis of the protocol in [4]. The cities are interconnected by highways across specific access points or ports and because of the regular traffic of highway mobile users in these highways, we assume that in each specific moment in time the probability of a highway mobile user being at a city's access port is p (in practice, these probabilities may differ between the various access ports and also vary with time, so we take, for analysis reasons, p to be a lower bound on these probabilities). For this model of ad-hoc mobile networks, we propose the following hierarchical routing protocol:

a) When some mobile host of the support is within the communication range of a sender, an underlying sensor subprotocol notifies the sender to give its messages to the support.

b) When the head of the support, performing a random walk on the motion graph of the city, arrives at the city's access port to the highway and it happens (with probability at least p) to meet there a mobile host leaving towards the highway, it delivers the messages to this mobile host. If there is no mobile host at the access port when the head of the support gets there, the support keeps moving randomly in the city until a subsequent successful (i.e. meeting a leaving to the highway mobile user) visit to the access point and the delivery of the messages to some mobile host moving towards the highway.

c) This mobile user, after having got the messages from the one city's support, moves (according to its regular movement on the highway) to the other city's access point, where again with probability at least p it meets the other city's support and delivers the messages to it.

d) Having received the messages from the mobile host at the access point, the support forwards them to the receiver host when meeting it during the random walk in the city.

We wish to emphasize here that although the analysis and the experiments described in the rest of this paper have been carried out for the basic case of two cities connected by a single highway, our protocol and its analysis can be easily extended to any number of cities interconnected across various access ports by a sparse network of highways. This is so because our approach is inherently modular in the sense that it actually assumes, for shaping the hierarchy of supports as a whole, a basic building block: a city and its access port to some highway. Thus, the changes incurred by adopting the protocol to any number of cities connected across various access ports by a sparse network of frequently used highways, are basically quantitative and easy to analyze, and do not affect the correctness and the essence of our approach.

Additionally, the lower bound on the probability p for meeting a mobile host heading towards the highway when a city's support reaches its access port is clearly a function of the number of mobile users moving on the highway, their speeds of movement and the length of the highway. Assuming appropriate values for these parameters, we concentrate here on constant values for the probability p.

5 Analysis of the HSRP

Before providing some formal analysis of the expected communication times achieved by our protocol, we give the following intuitive explanation of its superiority over the straight-forward extension of the protocol in [4] in such hierarchical ad-hoc mobile networks.

Remark that such a straightforward extension of the support idea would assume a single "snake-like" support for the whole network which, in order to receive messages from a sender mobile user in one city and deliver them to a receiver user in the other city, would have to necessarily pass through the highway interconnecting the two cities, an event that has clearly very small probability, Thus the expected time needed for such a passage would be extremely high leading to big communication times.

On the contrary, the hierarchical protocol proposed in this work needs only a constant expected number of visits of the support to the access port, because of the constant (and independent for various visits) probability of a successful visit (a visit when the support's head meets some highway mobile user) leading to a geometric distribution for the number of needed visits.

Remark also that the motions of the city mobile users, which are not members of the support, are determined by application protocols and that they are *independent* of the motion of the city's support (i.e. we exclude the case where the other city mobile users are deliberately trying to avoid Σ). Moreover, we assume that the mobile hosts of the network have sufficient power to support motion and communication.

In such cases, any particular mobile user will eventually meet some node of the support with probability 1. In fact, and by using the Borel-Cantelli Lemmas for infinite sequences of trials, given an unbounded period of (global) time (not necessarily known to the mobile stations) each user will meet the support *infinitely often* with probability 1 (since the events of meeting the support are mutually independent and the sum of their probabilities diverges). This fact guarantees correct delivery of a message onto its city's support Σ and, then, correct reception by a destination node when it subsequently meets its city's support.

We proceed by estimating the communication times achieved by our protocol.

The time needed for two mobile users in different cities to communicate is:

$$T_{total} = X + X_{\Sigma} + X_{AP} + T_{highway} + Y_{AP} + Y_{\Sigma} + Y$$

where X, Y represent the times (which are random variables) needed for a mobile user to meet its city's support, respectively, X_{Σ}, Y_{Σ} are the times for the messages to propagate within each city's support, respectively. X_{AP} and Y_{AP} are random variables representing the times needed for the randomly moving support's head of each city to deliver (respectively, receive) the messages to (respectively, from) the corresponding access port. $T_{highway}$ is the time needed for the mobile user on the highway to carry the messages from one access port to the other.

Time-efficiency of semi-compulsory protocols for ad-hoc networks is not possible to estimate without a scenario for the motion of the mobile users not in the support (i.e. the non-compulsory part). However, in a way similar to [4, 7], we propose an "on-the-average" analysis by assuming that the movement of each city mobile user *is a random walk on the corresponding city graph G*. We propose this kind of analysis as a necessary and interesting first step in the analysis of efficiency of any semi-compulsory or even non-compulsory protocol for ad-hoc mobile networks. In fact, the assumption that the mobile users are moving randomly (according to uniformly distributed changes in their directions and velocities, or according to the random waypoint mobility model, by picking random destinations) has been used in [8], [5].

We assume further that all random walks are *concurrent* and that there is a global time t, not necessarily known to the hosts. We are now able to 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∈V, the probability that s_{t+dt}=v, v∈V, is $p(u,v)$·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.

We denote E_μ [ ] the expectation for the chain started at time 0 from any vertex with distribution μ (e.g. the initial distribution of the Markov chain).

Let $T_i = min\{t \geq 0 : s_t = i\}$ be the first hitting time on state i (the first time that the mobile host visits vertex i).

To estimate the expected values of X and Y, we work (in a way similar to [4]) as follows:

(a) Note first that X, Y are, statistically, of the same distribution, under the assumption that u, v are randomly located (at the start) in the corresponding city graph G. Thus $E(X) = E(Y)$.

(b) We now replace the meeting time of u and Σ by a hitting time, using the following thought experiment: (b1) We fix the support Σ in an "average" place inside G. (b2) We then collapse Σ to a single node (by collapsing its nodes to one but keeping the incident edges). Let H be the resulting graph, σ the resulting node and $d(\sigma)$ its degree. (b3) We then estimate the hitting time of u to σ assuming u is somewhere in G, according to the *stationary distribution*, $\vec{\pi}$, of its walk, on H. We denote the expected value of this hitting time by $E_\pi T_\sigma^H$.

Thus, now $X + X_\Sigma = Y + Y_\Sigma = E_\pi T_\sigma^H + \mathcal{O}(k)$. Proceeding as in [1] we have (see a proof in [4])

Lemma 1 *([1]) For any node σ of any graph H in a continuous-time random walk*

$$E_\pi T_\sigma^H \leq \frac{\tau_2(1-\pi_\sigma)}{\pi_\sigma}$$

where π_σ is the (stationary) probability of the walk at node (state) σ and τ_2 is the relaxation time of the walk.

Note 2 *In the above bound, $\tau_2=\frac{1}{\lambda_2}$ where λ_2 is the second eigenvalue of the (symmetric) matrix S=$\{s_{i,j}\}$ where $s_{i,j}$=$\sqrt{\pi}_i\ p_{i,j}\ (\sqrt{\pi_i})^{-1}$ and P=$\{p_{i,j}\}$ is the transition matrix of the walk.*

It is a well-known fact (see e.g. [13]) that $\forall v \in V_H$, $\pi_v = \frac{d(v)}{2m'}$ where $m' = |E_H|$ is the number of the edges of H and $d(v)$ is the degree of v in H. Thus $\pi_\sigma = \frac{d(\sigma)}{2m'}$.

By estimating d(σ) and m' and remarking that the operation of locally collapsing a graph does not reduce its expansion capability and hence $\lambda_2^H \geq \lambda_2^G$.

Theorem 2

$$E(X) = E(Y) \leq \frac{1}{\lambda_2(G)}\Theta\left(\frac{n}{k}\right)$$

where n, k, $\lambda_2(G)$ are, respectively, the number of vertices of the motion graph of each city, the support size and the second eigenvalue of the adjacency matrix of the motion graph of each city.

Note 3 *The above upper bound is minimised when $k = \sqrt{\frac{2n}{\lambda_2(G)}}$, a fact also verified by our experiments.*

This analysis indicates the important fact that *only a small sized support Σ is needed in each city to achieve very efficient times for reaching the support*. This size (actually of order equal to the square root of the number of vertices in the motion graph) is also verified experimentally.

We now proceed with the analysis of the X_{AP} and Y_{AP} times. Remark that, X_{AP} and Y_{AP} have statistically the same distribution, thus their expected value, because of Lemma 1 and the fact that $\pi_{AP} = \frac{d_{AP}}{2m}$, is given by:

$$E(X_{AP}) = E(Y_{AP}) = \frac{1}{p}\frac{1}{\lambda_2(G)}\Theta\left(\frac{2m}{d_{AP}}\right)$$

(where m is the number of edges in the motion graph and d_{AP} is the degree of the access port vertex), since the expected number of visits of the support's head to an access port until a successful meeting with a highway mobile user is geometrically distributed with success probability p, and a visit of the support to the access port can be in fact viewed as a hitting time of a mobile user starting from a random point to the access port (since the support's head performs a random walk).

Now, we may naturally assume that the degree d_{AP} of the vertex corresponding to the access port is (because of its critical with respect to connectivity position in the network) at least $\overline{d}$, where $\overline{d} = \frac{2m}{n}$ is the average degree in the graph. Thus, we get

$$E(X_{AP}) = E(Y_{AP}) = \frac{1}{p}\frac{1}{\lambda_2(G)}\Theta(n)$$

Finally $T_{highway}$ is a function, as we have already said, of the highway traffic parameters and we consider this to be a given parameter of the protocol having constant size.

Thus, we finally get:

$$E(T_{total}) = \frac{1}{\lambda_2(G)}\left(\Theta\left(\frac{n}{k}\right) + \frac{1}{p}\Theta(n)\right) + 2\Theta(k)$$

which gives a linear average message delay $E(T_{total}) = O(n)$, where n is the number of vertices of the motion graph of each city.

6 Experimental Results and Algorithmic Engineering

6.1 Discussion on the Experiments

The experimental results have been used to evaluate, further investigate and comparatively study the performance of the Hierarchical Support Routing Protocol (HSRP) and the Original algorithm ([4]) in the new model of hierarchical ad-hoc networks.

In the experiments, we used one sender for generating and transmitting messages and one receiver for the destination of the messages, located at a different city. Both sender and receivers were part of the *city mobile users* group and were not allowed to move outside the area-borders of the dense subnetwork of their initial deployment (i.e. become members of the *highway mobile users* group). More crucially, we assumed that only one access port was available at each city, providing a direct connection between the two cities where the sender and the receiver where located. The experiments were carried out for different p and k for at least 500,000 exchanges regardless protocol generated traffic. In some cases (where k was near the optimum values stated in the analysis), we extended the message count to 800,000. In order to experiment on realistic cases, we used the $G_{n,p}$ model of random graphs. These graphs are obtained by sampling the edges of a complete graph of n nodes independently with probability p.

Two sets of experiments were carried out. The first set of experiments investigate the performance of the Original algorithm when applied to hierarchical ad-hoc mobile networks. We observe that communication between hosts located in different cities is successfully achieved, and that as the total number n of motion-graph nodes remains constant, as we increase the size k of Σ, the total message delay (i.e. $E(T_{total})$) is decreased. More importantly, the algorithm maintains the same basic behaviour described in [4], in the sense that $E(T_{total})$ initially decreases very fast with increasing k, while having a limiting behaviour of no further significant improvement when k crosses the threshold value indicated by the analysis in [4]. In figure 2 the curve of the performance of the Original algorithm is displayed.

The experiments indicate that, although the pattern of the Original algorithm's performance remains the same, it does not provide sufficient communication times even if the size k of the support is above the threshold value indicated in [4]. This implies the fact (also remarked analytically) that the average message delay is affected (in fact dominated) by the time (whose expectation is very high) required for

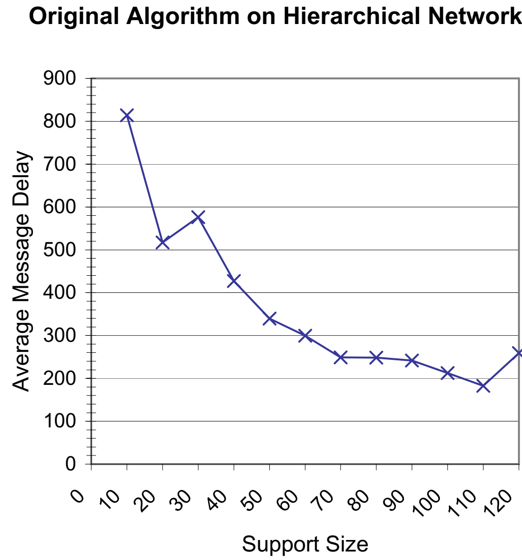

Figure 2. Average message delay over Support size k for two cities (n=1600) with one access port at each city and interconnected by one highway.

the support to reach the access port of the city and enter the highway in order to move to the other city and finally deliver the messages to their destination.

In the second set of experiments we evaluate the performance of the HSRP and we remark that $E(T_{total})$ only slightly depends on the actual size of the graph and the size of the support Σ but is mainly affected by the probability p measuring the frequency at which the highway mobile users arrive at the city's access port. This is also expressed throughout the theoretical analysis by the effect of the probability p on communication times. The curve of figure 3 clearly shows that for a fixed number of motion-graph nodes (e.g. n=1600) and fixed size of Σ (e.g. k=10) as the probability p increases, the overall average message delay drops. Actually, E(T_{total}) initially decreases very fast with increasing p, while having a limiting behaviour of no further significant improvement when p crosses a certain threshold value. Therefore, taking into account a possible amount of statistical error, the following has been experimentally validated:

$$\text{if } p_1 > p_2 \Rightarrow E_1(T_{total}) < E_2(T_{total})$$

Furthermore, the experimental results have been used to compare the average message delays for various support sizes k for a fixed number of motion-graph nodes (e.g. n=1600). It is shown that by increasing the support size k no significant performance is gained. An intuitive explana-

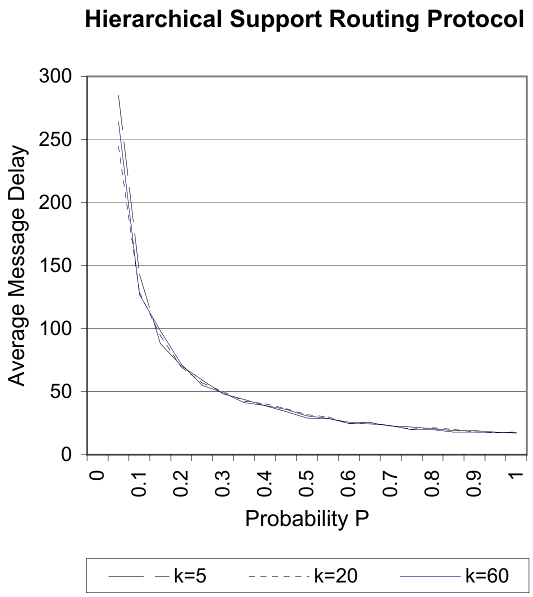

Figure 3. Average message delay over Probability p of meeting a highway mobile user when entering an access port for different support sizes k.

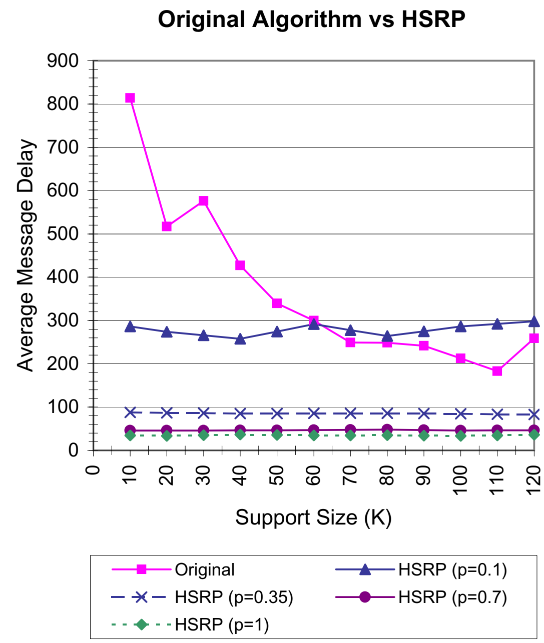

Figure 4. Original Algorithm versus HSRP for two cities (n=1600) with one access port at each city with varying support sizes k.

tion of the fact that in figure 3 all curves are almost identical is that the time for Σ to move to the access port of the city and encounter a highway mobile user dominates the overall message delay. If we take into account possible amount of statistical error that our experiments are inherently prone to, we can clearly experimentally conclude the following:

$$E(X_{AP}) >> E(X)$$

Finally, by combining the results of both sets of experiments we can compare the performance of the two routing algorithms. In figure 4 the overall average message delay of the algorithms is shown for a fixed number of motion-graph nodes (e.g. n=1600) per city and for a total of two cities. We remark that the communication times achieved by HSRP are indeed by far more efficient than those of the original algorithm even in the case of a small probability p. Moreover, figure 4 supports the claim that the effect of probability p over the communication times has a limiting behaviour of no further significant improvement when p crosses a certain threshold value.

More specifically, although the analysis cannot easily answer precisely the question of how frequent the movement of the highway mobile users must be in order to achieve high performance, experiments however indicate that a low probability p (infrequent movement of highway mobile users) suffices to achieve *efficient communication times* while higher probabilities (more frequent movement of highway mobile users) incurs only a slight improvement on the communication times. Actually, experiments imply that if p=0.35 the Hierarchical Support Routing Protocol achieves very efficient communication times even if a small support size k is chosen.

The experiments also indicate that the time required for the head of the support subnetwork to visit an access port and meet a highway mobile user dominates the overall communication times. More crucially, we remark that *a very small sized support Σ is needed in order to achieve very efficient communication times*. This size (actually of order bounded above by the square root of the number of vertices in the motion graph) is indeed verified experimentally.

Thus, taking into account a possible amount of statistical error, through the experimental validation process, the following can be claimed for the performance of HSRP:

$$\text{if } k_1 > k_2 \not\Rightarrow E_1(T_{total}) < E_2(T_{total})$$

Remark that, for the original protocol in [4], we have:

$$\text{if } k_1 > k_2 \Rightarrow E_1(T_{total}) < E_2(T_{total})$$

The experimental work on the proposed hierarchical adhoc mobile networks provides an important insight for finetuning of both the original algorithm and the hierarchical support routing protocol. It is well understood that the former does not achieve adequate communication times while the later performs with high efficiency.

6.2 Implementation of the Algorithm using LEDA

The algorithm was implemented into programs with the use of the Library of Efficient Data-types and Algorithms (LEDA, [12]). The library provides natural and elegant tools that made the transfer process from algorithms to programs very easy and fast. LEDA is ideally suited for rapid prototyping as summarized in the equation: $Algorithm + LEDA = Program$. Moreover, the data structures and algorithms in LEDA are efficient while the OOP enforced by C++ ensures code reusability.

To implement the Hierarchical Support Routing Protocol and modify the Original Algorithm so that it works on Hierarchical Ad-Hoc networks, we extended LEDA in order to support *the mobile host class*, the *message class* and the *transmission medium class*. Actually, the implementation used in [4] was the basis through which the new protocol was prototyped. By making the required modifications in specific parts of the original code, the reusability and ease of use of LEDA made the task of modifying the original protocol and implementing the new one very fast indeed.

6.3 Algorithmic Engineering

The extension of the original routing algorithm in [4] to support new hierarchical models of ad-hoc mobile networks imposes a number of critical issues. In order to evaluate, resolve and fine-tune such issues, the proposed protocol must be subjected to algorithmic engineering.

Throughout the model for hierarchical ad-hoc networks, the users have been distinguished between two well defined groups (city and highway mobile users). However, in reality, the users may freely move from one group to another (i.e. move to another city through the highways interconnecting the cities).

Instead of providing a subprotocol to determine and monitor the location of each mobile user, the hierarchical algorithm assumes that a recipient of a message can be located anywhere within the ad-hoc network. Therefore, each mobile support propagates messages to all neighbouring cities creating in such a way a number of multiple copies equal to the number of dense subnetworks (cities) that make up the hierarchical ad-hoc network. These copies will be stored at each city's support for a given period of time, sufficient to meet the recipient host if it lies within the areaborders of the city. This period can be set to be analogous to the cover time of each city's graph.

Furthermore, the cities may be interconnected through a number of access ports in order to form the hierarchical adhoc network. The number a of access ports of each city, the parameter p_{a_i} of each access port a_i and the interconnection topology of the cities affect the expected total average message delays in a way that seems difficult to be handled analytically and whose investigation may be facilitated by algorithmic engineering approaches.

7 Future Work

We intend to strengthen our results by providing a tighter analysis concerning the effect of probability p on the hitting times of the head of the support to the city's access port. We also wish to investigate the performance of the proposed algorithm on more complicated hierarchical ad-hoc networks and the corresponding effect on communication times of having many access ports per city.

References

[1] 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).

[2] 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).

[3] J. Broch, D. B. Johnson, and D. A. Maltz: The dynamic source routing protocol for mobile ad hoc networks. IETF, Internet Draft, draft-ietf-manet-dsr-01.txt, Dec. 1998. (1998).

[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, Sep. 2000. (WAE'00)(2000). Also see full paper at http://helios.cti.gr/adhoc/routing.html

[5] Z. J. Haas and M. R. Pearlman: The performance of a new routing protocol for the reconfigurable wireless networks. In Proc. ICC '98. (1998).

[6] Z. J. Haas and M. R. Pearlman: The zone routing protocol (ZRP) for ad hoc networks. IETF, Internet Draft, draft-zone-routing-protocol-02.txt, June 1999. (1999).

[7] 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). Also, short paper in Proc. 18th Annual Symposium on Principles of Distributed Computing (PODC'99) (1999).

[8] G. Holland and N. Vaidya: Analysis of TCP Performance over Mobile Ad Hoc Networks. In 5th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'99) (1999).

[9] T. Imielinski and H. F. Korth: Mobile Computing. Kluwer Academic Publishers. (1996).

[10] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek and M. Degermark: Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In 5th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'99) (1999).

[11] Y. Ko and N. H. Vaidya: Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In 4th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'98) (1998).

[12] K. Mehlhorn and S. Naher: LEDA: A Platform for Combinatorial and Geometric Computing. Cambridge University Press. (1999).

[13] R. Motwani and P. Raghavan: Randomized Algorithms. Cambridge University Press. (1995).

[14] 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).

[15] 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).

0-7695-0990-8/01/$10.00 (C) 2001 IEEE