Alcom-FT Technical Report Series
ALCOMFT-TR-01-61

Analysis and Experimental Evaluation of an Innovative and Efficient Routing Protocol 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. An ad-hoc mobile network is a collection of mobile hosts, with wireless communication capability, forming a temporary network without the aid of any established fixed infrastructure. In such a (dynamically changing) network it is not at all easy to avoid broadcasting (and flooding).
In this paper we propose, theoretically analyse and experimentally validate a new and efficient protocol for pairwise communication. The protocol exploits the *co-ordinated motion* of a *small* part of the network (i.e. it is a *semi-compulsory* protocol) in order to provide to various senders and receivers an efficient *support* for message passing. Our implementation platform is the LEDA system and we have tested the protocol for *three classes of graphs* (grids, random graphs and bipartite multi-stage graphs) each abstracting a different "motion topology".
Our theoretical analysis (based on properties of *random walks*) and our experimental measurements indicate that *only a small fraction* of the mobile stations are enough to be exploited by the *support* in order to achieve *very fast communication* between any pair of mobile stations.

1 Introduction

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 l 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 [3]), which delivers (if possible) a message from one mobile host to another, regardless of their position. This scheme, in the case of high mobility of

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

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 [4] 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. [2] present 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, [13]) is designed to minimise reaction to topological changes by localising routing-related messages to a small set of nodes near the change. [6] and [7] 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".

The innovative routing protocol that we propose here 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 [8] we use a graph theoretic concept where the movement of the mobile hosts in the three- dimensional space S is mapped to a graph G(V,E), |V|=n, |E|=e. In particular, 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. We expect that future research will incorporate geometry constraints into the subject). In particular, we first assume (as in [8]) 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 $\mathcal{V}(tc)$.

Our results: 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. The protocol is implemented in the LEDA environment by extending LEDA to support *the mobile host class, the message class* and *the transmission medium class*. Both our measurements and our theoretical analysis indicate that *only a small support is needed for very efficient communication*. Fine-tuning of our protocol is performed via a sequence of experiments for various graphs of motion. Our theoretical analysis draws from interesting properties of random walks on graphs and their meeting times. All our implementations are written in C++ and use

LEDA. The source codes are available at the following URL:

$$http://helios.cti.gr/adhoc/routing.html$$

2 The Protocol

2.1 The general scheme

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.

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.*

Our proposed scheme, 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.

2.2 The proposed implementation

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, $\forall i \in \{1,2,...,k\text{-}1\}$ 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 our experiments 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.

2.3 Protocol Correctness

We assume here that the motions of the mobile hosts of the network which are not members of the support are determined by application protocols and that they are *independent* of the motion of the support (i.e. we exclude the case where the other hosts 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 a case, describing most of the reasonable networks, any particular mobile host 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 station 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 guarantees correct delivery of a message onto the support Σ and, then, correct reception by a destination node when it subsequently meets the support.

2.4 Efficiency

Time-efficiency of semi-compulsory protocols for ad-hoc networks is not possible to estimate without a scenario for the motion of the mobile stations not in the support (the non-compulsory part). However, as in [8], we propose an "on-the-average" analysis by assuming that the movement of each mobile host $h \notin \Sigma$ *is a random walk on the 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.

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 host 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 $p(u,v)\cdot$dt where

$$p(u,v) = \begin{cases} \frac{1}{d(u)} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}, \text{ where } d(u) \text{ is the degree of vertex u.}$$

We denote $E_\mu[\,]$ 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). The total time needed for a message to arrive from a sending node u to a receiving node v is equal to:

$$T_{total} \;=\; X \;+\; T_\Sigma \;+\; Y \tag{1}$$

where:
X is the time for the sender node to reach a node of the support.
T_Σ is the time for the message to propagate inside the support. Clearly, $T_\Sigma = \mathcal{O}(k)$, i.e. linear in the support size.
Y is the time for the receiver to meet with a support node, after the propagation of the message inside the support.

To estimate the expected values of X and Y (they are random variables), we work 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 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*, $\boldsymbol{\pi}$, of its walk, on H. We denote the expected value of this hitting time by $E_\pi T_\sigma^H$.

Thus, now $E(T_{total}) = 2E_\pi T_\sigma^H + \mathcal{O}(k)$.

Note 1. The equation above amortises over meeting times of senders (or receivers) because it uses the stationary distribution of their walks for their position when they decide to send a message.

Note 2. Since the motion graph G is finite and connected, the continuous chain abstracting the random walk on it is automatically time-reversible.

Now, proceeding as in [1] we have (proof in the full paper, see [17])

Lemma 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 3. 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. Since S is symmetric, it is diagonalizable and the spectral theorem gives the following representation: S$=$UΛU^{-1} where U is orthonormal and Λ is a diagonal real matrix of diagonal entries $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_{n'}$, $n' = n - k + 1$ (where $n' = |V_H|$). These λ's are the eigenvalues of both P and S. In fact, λ_2 is an indication of the *expansion* of H and of the asymptotic rate of convergence to the stationary distribution, while relaxation time τ_2 is the corresponding interpretation measuring time.

It is a well-known fact (see e.g. [15]) 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 1.

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

Theorem 2. *The expected communication time of our scheme is bounded above by the formula*

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

Note 4. The above upper bound is minimised when k $= \sqrt{\frac{2n}{\lambda_2(G)}}$, a fact also verified by our experiments.

We remark that this upper bound is tight in the sense that by [1], Chapter 3,

$$E_\pi T_\sigma^H \geq \frac{(1-\pi_\sigma)^2}{q_\sigma\,\pi_\sigma}$$

where $q_\sigma = \sum_{j\neq\sigma} p_{\sigma j}$ in H_j is the total exit rate from σ in H. By using martingale type arguments we can show sharp concentration around the mean degree of σ.

3 Algorithmic Engineering

The fine-tuning of our protocol is a subject of algorithmic engineering since the theoretical analysis gives only bounds on the communication time. In particular, the analysis cannot easily answer the question of how much helpful it is for the head of Σ to remember (and avoid) past positions occupied by Σ. Experiments however indicate that limited memory (remembering a few past positions) incurs a slight improvement on the communication times, while bigger memory is not helpful at all.

More crucially, the analysis indicates the important fact that *only a small sized support Σ is needed to achieve very efficient communication times.* This size (actually of order equal to the square root of the number of vertices in the motion graph) is indeed verified experimentally.

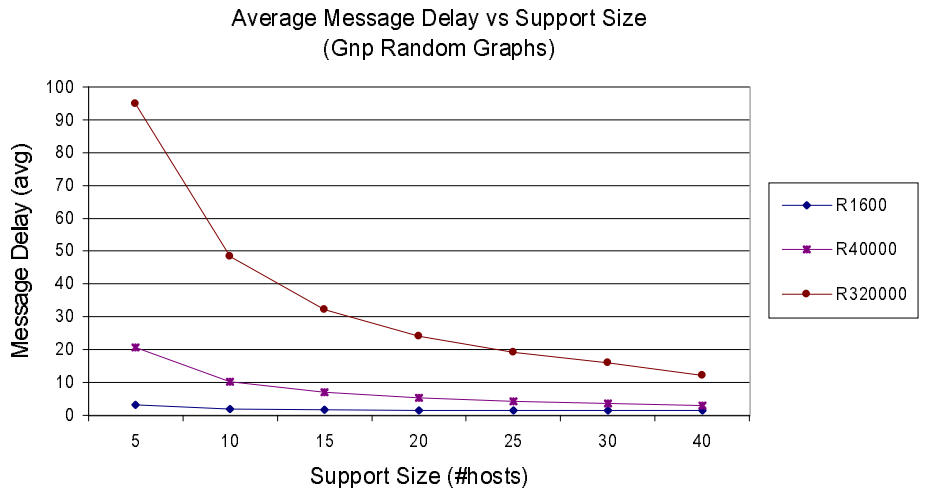

Fig. 1. Average message delay over Gnp types of motion graphs with different sizes n and varied support size k.

A number of experiments were carried out modelling the different possible situations regarding the geographical area covered by an ad hoc mobile network. We used one sender for generating and transmitting messages and one receiver for the destination of the messages and carried out the experiment with at least $100,000$ messages exchanges regarding protocol generated traffic. In some cases (such as the bipartite multi-stage graphs) we extended the message count to $150,000$.

For each message generated, we calculated the delay X (in terms of rounds) until the message has been transmitted to a member of Σ. Additionally, for each message received by the support, we logged the delay Y (in terms of rounds) until the message was transmitted to the receiver. We assumed that no extra delay was imposed by the mobile support hosts (i.e. $T_\Sigma = 0$). We then used the average delay over all measured fractions of message delays for each experiment in order to compare the results and discuss them further.

2D Grid Graphs. Our experimentation started with simple 2D grid graphs ($\sqrt{n}$ x $\sqrt{n}$). We used three different grid graphs with n$\in\{100,400,1600\}$ over different sizes of Σ with k$\in\{5, 10, 15, 20, 25, 30, 40, 60, 80, 100, 120, 160, 240, 320, 400, 480, 640\}$ (see figure 3).

$G_{n,p}$ Random Graphs. In order to experiment on more realistic cases, we used the $G_{n,p}$ model of random graphs, obtained by sampling the edges of a complete graph of n nodes independently with probability p. We used p=$1.05n\log n$ which is marginally above the threshold value for $G_{n,p}$ connectivity. The test cases included random graphs with n$\in\{100, 400, 1600, 3200, 40000, 320000\}$ and Σ size k$\in\{5, 10, 15, 20, 25, 30, 40\}$ (see figure 1).

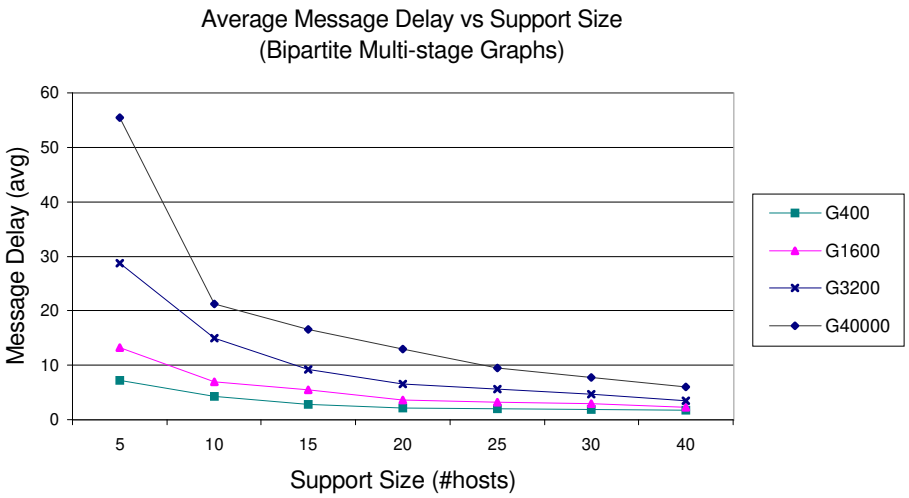

Fig. 2. Average message delay over bipartite multi-stage types of motion graphs with different sizes (n) and varied support size k.

Bipartite Multi-stage Graphs. The last set of experiments used the bipartite multi-stage graphs or series parallel graphs. In order to construct the graphs we used ξ stages, where ξ is a parameter. We used $p=\frac{n}{\xi}\log\frac{n}{\xi}$ which is the threshold value for connectivity of each pair of stages. The test cases included multi-stage graphs with 2, 3 or 4 stages and a total number of nodes n$\in\{100, 400, 1600, 3200, 40000\}$. The size of Σ we used was k$\in\{5, 10, 15, 20, 25, 30, 35, 40\}$ (see figure 2).

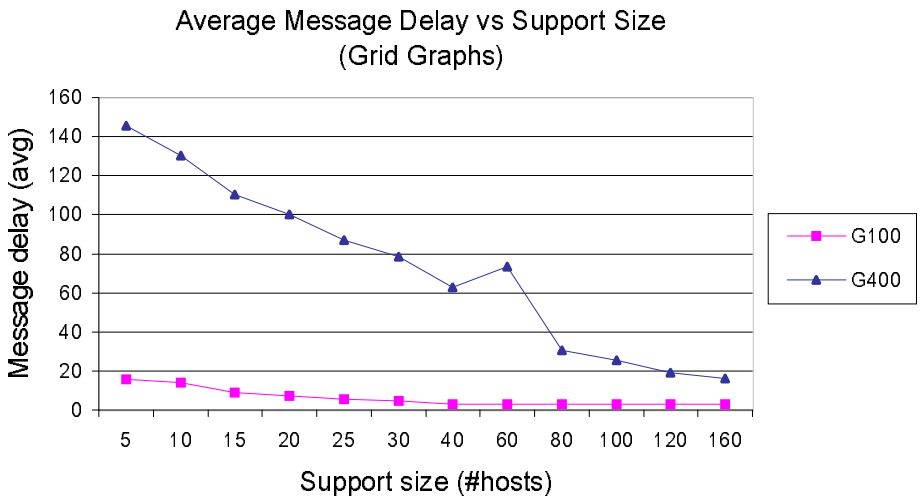

Fig. 3. Average message delay over grid types of motion graphs with different sizes (n) and varied support size k.

3.1 Discussion on the Experiments

The experimental results have been used to compare the average message delays with the upper bound provided by the theoretical analysis of the protocol. We observe 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. $\mathrm{E}(T_{total})$) is decreased. Actually, $\mathrm{E}(T_{total})$ initially decreases very fast with increasing k, while having a limiting behaviour of no further significant improvement when k crosses a certain threshold value. Therefore, taking into account a possible amount of statistical error, the following has been experimentally validated:

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

Throughout the experiments we used small and medium sized graphs, regarding the total number of nodes in the motion graph. Over the same type of graph (i.e. grid graph) we observed that the average message delay $\mathrm{E}(T_{total})$ increases as the total number of nodes increases. This can be clearly seen from figure 2 where the curves of the three different graph-sizes are displayed. It is safe to assume that for the same graph type and for a fixed-size Σ the overall message delay increases with the size of the motion graph. We further observe that $\mathrm{E}(T_{total})$ does not increase linearly but is affected by the expansion rate of the graph, or otherwise, as stated in the theoretical analysis, by the second eigenvalue of the matrix of the graph on communication times. If we take into account possible amount of statistical error that our experiments are inherently prone to, we can clearly conclude the following:

$$\text{if } n_1 \ > \ n_2 \ \Rightarrow \ E_1(T_{total}) \ > \ E_2(T_{total})$$

Furthermore, we remark that $\mathrm{E}(T_{total})$ does not only depend on the actual size of the graph and the size of the support Σ but additionally that it is directly affected by the type of the graph. This is expressed throughout the theoretical analysis by the effect of the second eigenvalue of the matrix of the graph on communication times. If we combine the experimental results of figures 1,2 and 3 for fixed number of motion-graph nodes (e.g. n=400) and fixed number of Σ size (e.g. k=40) we observe the following:

$$E_{grid}(T_{total}) \ > \ E_{multistage}(T_{total}) \ > \ E_{G_{n,p}}(T_{total})$$

which validates the corresponding results due to the analysis.

In section 2.4, we note that the upper bound on communication times $E(T_{total}) \leq \frac{2}{\lambda_2(G)}\,\Theta\left(\frac{n}{k}\right) + \Theta(k)$ is minimised when $k = \sqrt{\frac{2n}{\lambda_2(G)}}$. Although our experiments disregard the delay imposed by Σ (i.e. $\Theta(k) \to 0$) this does not significantly affect our measurements. Interestingly, the values of figures 1,2 and 3 imply that the analytical upper bound is not very tight. In particular, for the $G_{n,p}$ graphs we clearly observe that the average message delay drops below 3 rounds (which is a very low, almost negligible, delay) if Σ size is equal to 10; however, the above formula implies that this size should be higher, especially for the medium sized graphs. Actually, it is further observed that there exists a certain threshold value (as also implied by the theoretical analysis) for the size of Σ above which the overall message delay does not further significantly improve. The threshold size for the support validated experimentally is smaller than the threshold implied by the analytical bound.

The experimental work on the proposed routing protocol provides an important insight for fine-tuning of the original algorithm according to which the head of Σ remembers previous positions so that it does not re-visit them within a short period of time.

4 Future Work

We intend to strengthen our results by providing a tighter analysis on the concentration around the mean of the degree of the vertex corresponding to the collapsing of the support structure. We also wish to investigate alternative (not necessarily "snake-like") structures for the support and comparatively study the corresponding effect of support size on communication times.

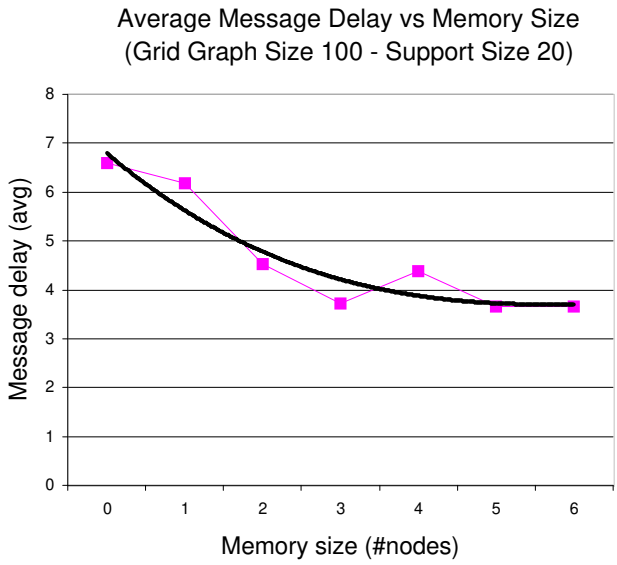

Fig. 4. The effect of variable memory size over the average message delay. The experiments use a grid type graph for the motion graph with a fixed number of nodes (n) and a fixed support size k=20.

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. 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).
3. 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).
4. 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).
5. B. Das, R. Sivakumar and V. Bharghavan: Routing in ad hoc networks using a virtual backbone. In Proc. IEEE (IC3N'97) (1997).
6. Z. J. Haas and M. R. Pearlman: The performance of a new routing protocol for the reconfigurable wireless networks. In Proc. ICC '98. (1998).
7. 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).
8. 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).
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 Proc. Mobile Computing (MOBICOM'99) (1999).
11. Y. Ko and N. H. Vaidya: Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In Proc. Mobile Computing (MOBICOM'98). (1998).
12. K. Mehlhorn and S. Naher: LEDA: A Platform for Combinatorial and Geometric Computing. Cambridge University Press. (1999).
13. 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).
14. G. Aggelou and R. Tafazolli.: RDMAR: A Bandwidth-efficient Routing Protocol for Mobile Ad Hoc Networks. In Proc. 2nd ACM International Worksh. on Wireless Mobile Multimedia (WoWMom'99) (1999).
15. R. Motwani and P. Raghavan: Randomized Algorithms. Cambridge University Press. (1995).

16. E. M. Royer and C. K. Toh: A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. (1999).
17. I. Chatzigiannakis, S. Nikoletseas and P. Spirakis: Analysis and Experimental Evaluation of an Innovative and Efficient Routing Protocol for Ad-hoc Mobile Networks (full paper). CTI Technical Report. Aug. 2000. http://helios.cti.gr/adhoc/routing.html (2000).