

Encounter-Based Message Propagation in Mobile Ad-Hoc Networks

D.E. Cooper^{*}, P. Ezhilchelvan I. Mitrani

*Newcastle University, School of Computing Science, Newcastle upon Tyne, NE1
7RU, UK*

Abstract

A family of message propagation protocols for highly mobile ad-hoc networks is defined, and is studied analytically and by simulation. The coverage of a message (the fraction of nodes that receive it), can be made arbitrarily close to 1, at a moderate cost of extra message traffic. Under certain simplifying assumptions, it is shown that a high coverage is achieved by making a total of $O(n \ln n)$ broadcasts, where n is the number of nodes, and the time to propagate a message is $O(\ln n)$. Mechanisms for making the protocols more efficient, by reducing the number of redundant transmissions without affecting the achievable coverage, are presented and evaluated. The generalisation to multiple broadcasts proceeding in parallel is derived. Finally, an attempt to validate the proposals in the context of a real-life mobile ad-hoc network is described. Results of this validation exercise and simulations point out that the proposed protocols can cost-effectively achieve high-coverage in networks of varying degrees of node densities and mobilities.

We acknowledge the financial support from the UK EPSRC Projects “Networked Computing in Inter-organisation Settings” (EP/D037743/1) and “Protocols for Ad-hoc Collaborative Environments” (GR/S02082/01).

Key words: Ad-hoc networks, message propagation, broadcasting, encounters.

1 Introduction

Recent advances in the technologies of mobile devices and wireless communication have given rise to an increasingly popular form of networking, called mobile

^{*} Corresponding author.

Email addresses: d.e.cooper@ncl.ac.uk (D.E. Cooper),
paul.ezhilchelvan@ncl.ac.uk (P. Ezhilchelvan), isi.mitrani@ncl.ac.uk (I. Mitrani).

ad-hoc networking. A mobile ad-hoc network (MANET) consists of small, versatile and powerful mobile computing devices (nodes). It is typically formed at short notice and does not make use of any fixed networking infrastructure. A distinguishing feature of a MANET is that the nodes are not just the sources of message traffic but also engage in forwarding messages to final destinations; given that the nodes can be highly mobile, a MANET is a dynamic network characterised by frequent and hard-to-predict topological changes.

An application of a mobile network usually involves user collaboration towards achieving a common goal, in situations where access to base stations is unavailable or unreliable (e.g., command and control or disaster relief). The success of such collaborative undertakings depends to a large extent on the provision of reliable, fast and economic multicast. That is, a message originating at any node should reach all other nodes within a reasonably short period of time and without consuming much of the network resources. Unfortunately, both the nature of the devices and their mobility imply that these are conflicting objectives whose achievement cannot normally be guaranteed. One is therefore obliged to trade off coverage (i.e., the fraction of nodes that receive a message) against message traffic and delays.

Our objective is to devise and evaluate message propagation protocols which achieve very high coverage without paying too high a price in terms of propagation delays. We also introduce mechanisms aimed at reducing the price paid in terms of message traffic.

Existing work in this area has concentrated on different aspects of the above trade-offs, with varying degree of success depending on the nature of the network. For example, several protocols (see [9,25,10,3]) aim to minimise the number of broadcasts by using state information about the network topology. When the degree of mobility is low, these protocols perform well, but when it is high, the network state information can become out-of-date quickly and the coverage achieved can be poor [23,28].

A topology-independent and stateless protocol that seems to work better in highly mobile networks is ‘flooding’; see Ho et al [14]. In its simplest form, every node broadcasts every message once, either immediately upon receipt or after a random interval. The coverage achieved by flooding depends considerably on the mobility pattern and on the ‘density’ of nodes (usually defined as the average number of nodes within a disc of radius equal to the wireless range). When the density is high, the coverage is potentially high, but flooding causes ‘broadcast storms’, with their attendant problems of wasted network resources and possibility of collisions. To counter these problems, Ni et al [21] have proposed and studied a number of optimisations, some of which have applications in our protocols.

On the other hand, low density or particular mobility patterns render the network liable to ‘partitioning’ (Hahner et al, [12]). For such networks the flooding coverage tends to be poor (Khelil et al, [17], Obraczka et al, [22]).

Modifications of the flooding protocol, such as ‘Hypergossiping’ [17,16] and ‘Adaptive-Flooding’ [27] aim to improve coverage by allowing nodes to make more than one broadcast of a message during the life-time of that message. However, the efficacy of those refinements depends heavily on being able to select the life-time duration appropriate to the node speed. The speed is not easy to determine without providing the nodes with special equipment, and even when determined, it is not clear what is the best choice for fixing the life-time duration. Appropriate setting of the life-time parameter is difficult and requires careful calibration of system behaviour [17]. Our approach eliminates these problems without sacrificing coverage. We propose, and study, a family of protocols which preserve the topology-independent nature of flooding, while being able to achieve coverage levels arbitrarily close to 1, for any node density. Of course a specific high coverage cannot be guaranteed in any given instance, but can be expected with high probability. These protocols are based on a notion of ‘encounter’, and are controlled by an ‘encounter threshold’ parameter. The cost paid for a high coverage is an increase in the message traffic, since messages are broadcast more than once by each node. Under certain simplifying assumptions, it is shown that to achieve a coverage close to 1 in a network with n nodes, the total average number of broadcasts per message is on the order of $O(n \ln n)$. This is a moderate increase on the $O(n)$ broadcasts carried out in flooding. The propagation time of a message is on the order of $O(\ln n)$. Thus, these protocols are particularly recommended for low density, highly mobile networks where reasonable delays can be tolerated.

At higher densities, our protocols, in their simple form, exacerbate the broadcast storm problem. To remedy this, we use existing and new optimisations which are efficient in reducing the number of broadcasts, without reducing coverage significantly and without requiring topology information.

Various aspects of the protocols’ performance are examined by simulation. We have also conducted an experiment (similar to the one reported in Hui et al [15]), where performance measurements were taken from a real-world ad-hoc network.

The model, and the message propagation protocols, are described in section 2. Some analytical results concerning the propagation time and the number of broadcasts are obtained in section 3. The problem of improving the efficiency of the protocol by reducing the number of redundant broadcasts is addressed in section 4. In section 5, the protocol is generalised in order to handle the propagation of multiple messages in parallel. The outcomes of a number of simulation experiments with the basic, improved and generalised protocols, and

with two different mobility patterns, are presented in section 6. The validation exercise involving a real-life ad-hoc network is described in section 7. Section 8 summarises the results obtained and outlines avenues of further enquiry.

2 The model

The system under consideration consists of n mobile nodes which move within a given terrain. The nodes communicate with each other using wireless technology, but without any fixed network infrastructure support. That is, the nodes themselves are the sources as well as the forwarders of the message traffic, and thus form a mobile ad-hoc network. Each node has a unique identifier (MAC or IP address). It is assumed that nodes do not run out of power and do not fail; however, due to their mobility, they may become disconnected, and reconnected, as they move out of and into each other's wireless range. Thus, the structure of the network can change with time in an unpredictable manner. For simplicity, assume that the wireless ranges of all nodes are equal and remain constant during the period of interest.

The movement of each node is governed by some 'mobility pattern', which controls its current speed and direction. It is assumed that the n nodes are statistically identical, i.e. the rules of their mobility patterns are the same, and any random variables involved have the same distributions for all nodes.

We shall define a protocol whose principal objective is to deliver a message, originating at any node, to all other nodes with high probability. A secondary objective is to minimise, as far as possible, the memory requirements at each node. In fact, what will be defined is not a single protocol, but a family of protocols depending on an integer parameter, τ .

Node i ($i = 1, 2, \dots, n$) advertises its presence by broadcasting, at regular intervals, a signal carrying its identifier and saying, essentially, 'hello, this is node i '. It also listens for similar signals from other nodes and maintains a list, $\{j_1, j_2, \dots, j_k\}$, of the nodes, other than itself, that it can hear. That list is called the 'current neighbourhood' of node i . At any moment in time, any current neighbourhood may be empty, or it may contain any number of other nodes.

The current neighbourhood of node i changes when a node which was in it, say j_1 , moves out of range, or when a node which was not in it, say j_{k+1} , moves into range. The latter event is called an 'encounter'; that is, node i is said to encounter node j_{k+1} . Note that, since 'hello' signals are not assumed to be synchronised among the nodes, if node i encounters node j , node j does not necessarily encounter node i at the same time. Also note that, if node j leaves

the current neighbourhood of node i and at some later point enters it again, then that entry constitutes an encounter. Nodes do not maintain a history of their current neighbourhoods, in order to keep their memory requirements low.

Now consider a message propagation protocol where each node behaves as follows:

1. Upon receiving or originating a new message, m , store it, together with an associated counter, $c(m)$, which is set to zero. Add the sending node to the current neighbourhood, unless already present. If the current neighbourhood contains nodes other than the sending one, broadcast m and increment $c(m)$ by 1.
2. At every encounter thereafter, if $c(m) \leq \tau$, broadcast m and increment $c(m)$ by 1.
3. When $c(m) = \tau + 1$, remove m from memory (but keep its sequence number in order to remember that it has been handled).

Thus, every node receiving a message broadcasts it at $\tau + 1$ consecutive encounters (one of which may be the message arrival), and then discards it. There are no acknowledgements. The integer τ is called the ‘encounter threshold’. The above protocol will be referred to as ‘Encounter Gossip’, and will be denoted $EG(\tau)$, in order to make explicit the dependence on encounter threshold τ .

When $\tau = 0$, the $EG(0)$ protocol behaves like flooding (except that the broadcast is delayed until the next encounter if the current neighbourhood contains only the sender). At the other extreme, if $\tau = \infty$, we have an $EG(\infty)$ protocol whereby messages are kept forever and broadcast at every encounter. Assuming that the mobility pattern is such that every node eventually encounters every other node, $EG(\infty)$ achieves coverage 1. Of course, $EG(\infty)$ is not a practical option, but we shall see in Section 3 that it can provide some useful insights.

It should be pointed out that $EG(\tau)$ trades memory capacity and probability of reaching all nodes against message traffic. Because past histories are not kept and exchanged, messages may be sent again to nodes who have already received them. By increasing the value of τ , the coverage can be made to approach 1, at the cost of having to store more messages for longer periods, and making more broadcasts.

The performance measures of interest are:

- (i) The average response time of $EG(\tau)$, defined as the interval between the arrival (origin) of a message and the moment when no node can propagate it further.
- (ii) The average propagation time of a message, defined as the interval between its arrival and the moment when either all nodes have received it, or no node

- can propagate it further.
- (iii) The coverage of a message, i.e. the fraction of nodes that have received it by the end of its propagation time.

All of these performance measures are stated in terms of averages. However, the simulation results reported in Section 4 provide some indication of the corresponding variances, by repeating each experiment 10 times with different random number streams. For example, observing a coverage of 1 implies that *all 10 runs* achieved a coverage of 1.

It is important to be able to choose the value of τ so as to achieve high coverage, without unduly increasing the response and propagation times. This question will be addressed in the following sections.

3 Analytical approximation

In this section, we concentrate on evaluating the ability of $EG(\tau)$ to achieve high coverage. In order to make the model tractable, we assume the following:

- The overheads of collision resolution are negligible.
- Hello signals are sent and monitored at the MAC level; the information necessary to maintain the neighbourhood list is obtained at no extra cost to the higher level protocol.
- Encounters last long enough for a message to be received, i.e. the processing and propagation times of hello and broadcast messages are small enough for the encountered node to remain in the range of the encountering node.

These assumptions will not be required in the simulation experiments.

Consider an idealised system with n mobile nodes who never cease to propagate the messages they receive (∞ -propagation). Let T be the random variable representing a message propagation time, i.e., the interval between the origin of a message at some node, and the first instant thereafter at which all nodes have received it. If messages are not discarded, and every node eventually encounters every other node, T is finite with probability 1. It is then of interest to estimate its average value, $E(T)$. That quantity will also be used in choosing a suitable value for τ , when designing a practicable $EG(\tau)$ protocol.

An estimate for $E(T)$ will be obtained under the following simplifying assumptions:

- (a) Each node experiences encounters at intervals which are exponentially distributed with mean ξ .
- (b) At each encounter, a node meets one other node.

- (c) The node encountered is equally likely to be any of the other nodes; that is, the probability that node i will next encounter node j , $j \neq i$, is equal to $1/(n - 1)$, regardless of past history.

Assumption (a) can be justified by remarking that the interval until the next encounter experienced by a given node — say node 1 — is the smallest of the intervals until its next encounters with node 2, node 3, \dots , node n . Some of these intervals may in fact be of length 0 with a positive probability. Nevertheless, it is reasonable (e.g., see [20]) to assume that the interval until the first of many random occurrences is approximately exponentially distributed. The value of ξ depends on the density of nodes, on the speed with which they move, and on the mobility pattern. It may be difficult to determine ξ analytically, but in practice it can be estimated by monitoring the system and taking measurements.

Assumption (b) is deliberately pessimistic, in order to give the estimate the character of an upper bound. If a node encounters more than one other node at the same time, then the propagation will proceed faster. In fact, it will be seen in the experiments that at high densities this assumption is *very* pessimistic.

Assumption (c) is loosely based on the fact that all nodes are statistically identical, and move independently of each other. If the starting positions of the nodes are uniformly distributed, the assumption is justifiable at the first encounter, although it may well be violated in subsequent ones. However, this assumption provides the simplification necessary for analytical tractability. Its effect on the performance measures will be evaluated in the simulation experiments.

Let $X = \{X(t); t \geq 0\}$ be the Markov process whose state at any given time is the number of nodes that have already received the message. The initial state of X is $X(0) = 1$ (only the originating node has received it; again, this is a pessimistic simplification since the neighbourhood of the originating node may in fact contain other nodes). The random variable T is the first passage time of X from state 1 to state n .

Suppose that X is in state k , i.e. k nodes have received the message and $n - k$ have not. If any of the former k nodes encounters any of the latter $n - k$, the process will jump to state $k + 1$. Since each node experiences encounters at rate $1/\xi$, and the probability of encountering any other node is $1/(n - 1)$, the transition rate of X from state k to state $k + 1$, $r_{k,k+1}$, is equal to

$$r_k = \left[\frac{k}{\xi} \right] \left[\frac{n - k}{n - 1} \right]. \quad (1)$$

In other words, the average time that X remains in state k is

$$\frac{1}{r_k} = \frac{(n-1)\xi}{k(n-k)}. \quad (2)$$

Hence, the average first passage time from state 1 to state n is given by

$$E(T) = (n-1)\xi \sum_{k=1}^{n-1} \frac{1}{k(n-k)}. \quad (3)$$

This last expression can be simplified by rewriting the terms under the summation sign in the form

$$\frac{1}{k(n-k)} = \frac{1}{n} \left[\frac{1}{k} + \frac{1}{n-k} \right].$$

The two resulting sums are in fact identical. Therefore,

$$E(T) = \frac{2(n-1)\xi}{n} \sum_{k=1}^{n-1} \frac{1}{k} = \frac{2(n-1)\xi H_{n-1}}{n}, \quad (4)$$

where H_n is the n th harmonic number. When n is large, the latter is approximately equal to

$$H_n \approx \ln n + \gamma,$$

where $\gamma = 0.5772\dots$ is Euler-Mascheroni's number. Also, when n is large, $(n-1)/n \approx 1$ and $\ln(n-1) \approx \ln n$.

We have thus arrived at the following estimate, valid under assumptions (a), (b) and (c):

Proposition 1 *In a large mobile network where messages are not discarded, the average propagation period for a message is approximately equal to*

$$E(T) \approx 2\xi(\ln n + \gamma). \quad (5)$$

An immediate corollary of Proposition 1 is that, during the propagation period T , the originating node experiences an average of $2(\ln n + \gamma)$ encounters. Other nodes, who receive the message later on, tend to experience fewer encounters. Thus, choosing the encounter threshold, τ , to have the value

$$\tau = 2\lceil \ln n + \gamma \rceil, \quad (6)$$

should ensure that, when the protocol terminates, most nodes will have received the message. This suggestion will be tested experimentally.

Note 1. An attractive aspect of equation (6) is that the only parameter appearing in it is the number of nodes, n . If the assumptions (a), (b) and (c) are satisfied reasonably well, then the mobility pattern and the node density do not matter. However, for some mobility patterns those assumptions, and in particular (c), may be difficult to satisfy.

Note 2. Since, under the $EG(\tau)$ protocol, every node that receives a message broadcasts it $\tau + 1$ times, the total number of broadcasts per message is on the order of $O(n(\tau + 1))$. Hence, if τ is chosen according to (6), the total number of broadcasts per message is on the order of $O(n \ln n)$.

4 Protocol Improvements

Encounter propagation involves a trade-off between coverage and propagation overheads. The larger the value of τ , the higher the coverage achieved, but also the higher the number of redundant broadcasts (a broadcast is redundant if it does not enlarge the set of nodes that have already received the message). Each broadcast consumes power, shortens the battery operative period and, by increasing channel traffic, increases the likelihood of collisions. It is therefore important to keep the number of broadcasts to a minimum.

Since every non-redundant broadcast adds at least one node to those that have received the message, there can be at most $n - 1$ non-redundant broadcasts associated with a given message. On the other hand, if all nodes receive the message, a total of $n(\tau + 1)$ broadcasts are made under $EG(\tau)$. Therefore, when the coverage is 1, at least $n\tau + 1$ broadcasts are redundant. The aim is to reduce that number while still achieving a high coverage.

The general idea of the approaches proposed here is to suppress a broadcast of a message if that broadcast is judged to have little additional effect on its propagation, and to treat the suppressed broadcast as though it had been carried out. Consequently, a node may end up doing fewer than $\tau + 1$ broadcasts while the coverage remains largely unaffected.

The broadcast-suppression mechanisms that are introduced can be described as either ‘state-based’ or ‘history-based’. The state-based approach causes node i to update its broadcast counter without making a broadcast when certain events occur. In the history-based approach, each node maintains a local list of nodes that are known to have received the message. When node i experiences an encounter, it suppresses its broadcast if the encountered node is already in

that local list.

One state-based technique is the Random Assessment Delay (RAD). Having decided to broadcast a message as a result of an encounter (not as an originator), a node waits for a random period of time, called the ‘RAD interval’. An encounter which occurs during a RAD interval does not generate a new RAD interval. If, during a RAD interval, a node hears another broadcast of the same message, then the planned broadcast is suppressed. This is an existing methodology which has been applied in conjunction with other protocols (see [28,21]). The rationale is that if several nodes in a given neighbourhood are in possession of the message and decide to broadcast it, one of them will do so first (the one with the shortest RAD interval), and then the others can keep quiet.

Remember that under Encounter Propagation a node ceases to transmit message m when the number of encounters, recorded in $c(m)$, reaches the value τ . Now, the addition of RAD may affect the way $c(m)$ is incremented. During a RAD interval associated with message m , the node counts the number of transmissions of m that it hears. Denote that number by r . If $r = 0$ at the end of the RAD interval, then the node transmits m and increments $c(m)$ by 1. If $r > 0$, there are two possibilities:

1. Do not transmit m and increase $c(m)$ by 1;
2. Do not transmit m and increase $c(m)$ by r .

Both of these policies were studied and it was found that their performance is similar, with policy 2 performing marginally better than policy 1 (in the sense that it achieves a slightly larger reduction in redundant broadcasts, without significant adverse effect on the coverage). So, in order to avoid duplication, policy 1 will not be considered further; the RAD results presented in Section 5 concern policy 2 only.

Suppose that node i has already received and perhaps broadcast message m , and now hears it broadcast by a node j present in its current neighbourhood (i.e., node i is not experiencing an encounter). This can happen because node j has an encounter with another node, k , which is outside i 's neighbourhood. Node i thereby learns that a node carrying the same message, outside the current neighbourhood, is passing within 2 radii of it. One can argue that, in the light of this information, node i should not proceed to make the number of broadcasts required by $EG(\tau)$, but should increment its counter $c(m)$ by an amount reflecting the density of nodes in this region. It is proposed, therefore, that on hearing a broadcast of a message already held, by a node already present in the neighbourhood, node i should increment $c(m)$ by a fraction, α , of the number of nodes in its current neighbourhood (truncated to an integer).

Incrementing $c(m)$ causes the number of future broadcasts to be reduced. This

policy will be referred to as α -reduction. To determine a good value for α , we experimented by varying α in the range (0,1), for different parameter sets. The value $\alpha = 0.39$ was found to perform well over a wide range of parameters. That number happens to be (approximately) the minimum area of intersection of two equal circles whose centers are within each other's radii, as a fraction of the area of one of them.

The history-based approach requires node i to maintain a local cache where it can store the *ids* of all nodes it has encountered since first receiving (or originating) message m . It would be reasonable to assume that all nodes on that list have received m (that assumption is wrong only if a broadcast message failed to reach some of the neighbouring nodes). Therefore, if node i encounters node j and discovers that j is already on its list of receivers, it can suppress the broadcast required by EG(τ) and increment $c(m)$ by 1. This policy will be referred to as *encounter history* reduction, or EH.

A more comprehensive history of nodes that are presumed to have received a given message can be maintained by passing cache information at encounter events. Any node broadcasting m can piggy-back its current list of *ids* onto the message being broadcast; the receiving nodes would then merge that list with their own. Thus, the current list kept by node i contains not only the nodes to whom i has broadcast m , but also nodes about which it has been told that they have received m . Again, if node i encounters node j and finds that j is already on its list of receivers, it suppresses its broadcast and increments $c(m)$ by 1.

This policy is called *propagation history* reduction, or PH. For PH to be scalable, the size of the piggy-backed information needs to be kept small. This can be achieved by limiting the number, k , of *ids* that are piggy-backing onto a message. If, at the time of a broadcast, a node's cache contains more than k *ids*, k of them are selected for inclusion in m . The selection criterion may be, for example, FIFO, LIFO, or random.

Clearly, there is a trade-off between the size, k , of the list that may be attached onto a message, and the efficiency of the propagation protocol. The larger the value of k , the fewer redundant messages will be sent, but also the larger each broadcast will be, and hence the smaller the fraction of 'essential' information transmitted per broadcast. That trade-off is not studied to any great extent here. The experiments in Section 5 assume that $k = n$, and thus provide an upper bound on the achievable reduction of redundant broadcasts.

Another method for reducing redundant transmissions in networks liable to partitioning is for the encountering nodes to first exchange message histories and then decide which, if any, messages to transmit [17,26]. This trades off the benefit of reducing redundant message transmissions against the overhead of

the history exchange and the likelihood that, at high relative speeds, the nodes will move out of range during it. In relying only on propagation histories, we avoid that overhead. A limited experimental comparison of the two approaches will be presented in section 6.

5 Multiple messages

In this section we address some of the issues that arise when more than one propagation, of messages originating at different nodes at different times, may overlap. To achieve such multiple propagations, each node must maintain a buffer of all messages that it has received, together with the corresponding counts indicating how many times each message has been broadcast. A simple generalisation of the $EG(\tau)$ protocol, where each node can keep track of up to M messages in the process of propagation, has the following structure.

- (1) Upon receiving or originating a new message, m_i , if there is room in the local buffer, store m_i in it, together with an associated counter, $c(m_i)$; the latter is set to zero. Add the sending node to the current neighbourhood, unless already present. If the current neighbourhood contains nodes other than the sending one, broadcast m_i and increment $c(m_i)$ by 1. If the buffer is full when m_i arrives, it is rejected.
- (2) At every encounter thereafter, for each buffered message, m_i , if $c(m_i) \leq \tau$, broadcast m_i and increment $c(m_i)$ by 1.
- (3) When $c(m_i) = \tau + 1$, remove m_i from the buffer (but keep its sequence number in order to remember that it has been handled).

This protocol will be referred to as ‘Simple Encounter Gossip with Multiple Messages’ and will be denoted by $SEG(\tau, M)$. For the purposes of this study, M is chosen sufficiently large so that messages are never rejected.

Broadcasting several messages one after another may take quite a long time. If, during that period, nodes are likely to join or leave the current neighbourhood, the efficiency of $SEG(\tau, M)$ can be seriously impaired. Suppose, for example, that node 1 encounters node 2 and starts broadcasting k messages, m_1, m_2, \dots, m_k , that are currently in its buffer. During the transmission of m_i , node 2 leaves the neighbourhood. Then the transmissions of $m_{i+1}, m_{i+2}, \dots, m_k$ are unnecessary and it would be better to suppress them.

Alternatively, suppose that during the transmission of m_i , node 3 joins the neighbourhood. This is a new encounter, so $SEG(\tau, M)$ schedules another broadcast of all k messages (assuming that their counters have not exceeded τ), to begin as soon as the current schedule completes. The resulting retransmission of messages m_1, m_2, \dots, m_i is necessary, because node 3 has not received

them. However, the repeated transmission of $m_{i+1}, m_{i+2}, \dots, m_k$ is unnecessary; besides wasting power, it hastens the termination of the protocol and hence reduces the probability of full coverage.

Another problem caused by premature departures is that messages near the tail of a *FIFO* buffer are less likely to be propagated successfully (because the target node leaves before their turn comes), than those near the head. However, that problem is easily cured by using a ‘circular’ buffer instead of a *FIFO* one. An integer indicating the index of the currently transmitted message is incremented by 1 (modulo M) after each transmission. A newly received or locally generated message is inserted immediately to the left of the current index (modulo M).

In order to eliminate the redundant transmissions which may result from arrivals or departures during broadcasts, each node maintains, in addition to the list of nodes in the current neighbourhood, a sublist of ‘target’ nodes: these are nodes that have still not received all the messages in the circular buffer. For each target node, the index of the ‘last-due’ message, i.e. the last message that it needs to receive, is recorded. Step 2 of the $SEG(\tau, M)$ protocol is modified as follows:

2. At every encounter thereafter, add the newly encountered nodes to the target list, with last-due index equal to the current index - 1 (modulo M). Then, for each buffered message, m_i , if $c(m_i) \leq \tau$ and the target list is not empty: broadcast m_i ; increment $c(m_i)$ by 1; remove from the target list all nodes whose last-due index is equal to i , as well as all nodes that have meanwhile left the neighbourhood.

A node is deemed to have left the neighbourhood if no beacon has been received from it during a given interval of time. The latter is normally chosen to be larger than the inter-beacon interval, in order to include possible network delays.

The modified protocol will be referred to as ‘Encounter Gossip with Multiple Messages’ and will be denoted by $EG(\tau, M)$.

6 Simulation results

A number of simulation experiments were carried out, aimed at evaluating the effect of various parameters on the performance of the protocols described in the previous sections. The following factors were kept fixed:

The terrain is a square of dimensions $(1000\text{ m}) \times (1000\text{ m})$. The number of

nodes is kept fixed at $n = 64$. The node density (defined as the average number of nodes within a circle of radius equal to the wireless range) is varied by altering the wireless range. Values for the density used range from 0.5 to 6.5.

The interval between ‘hello’ signals for each node is 250 *ms*.

Two mobility patterns are examined: ‘Random Waypoint’ and ‘Manhattan Grid’. Under Random Waypoint, the nodes are initially distributed uniformly on the square; thereafter, each node chooses a random destination (also uniformly distributed on the square) and moves towards it at a given speed; upon reaching the destination, the node pauses for a given interval (1 *ms* in our case), selects a new random destination and so on.

The Manhattan Grid mobility pattern assumes that the area is covered by a square grid of ‘North-South’ and ‘East-West’ paths, at 40 *m* spacing. Initially, nodes are distributed regularly at the first n intersections of the grid. Thereafter, they move along the paths at a fixed speed. Whenever a node reaches an intersection, it chooses one of the four available directions with equal probability (at the edges of the grid the number of possible directions is reduced appropriately). We have used an implementation of the Manhattan Grid mobility pattern provided by the University of Oregon’s Network Research Group [24]. The first 1000 seconds of mobility are discarded, in order to remove initial bias.

The speed, node density and encounter threshold were varied and the performance measures — average response time, average propagation time and coverage — were evaluated. Each run starts at time 0 with a message originating at node 1, and terminates when no node can propagate the message further. For each set of parameter values, the simulation ran 25 times, with different random number seeds, and the performance observations were averaged.

We begin by evaluating the properties of the basic protocol $EG(\tau)$ (section 3). Figure 1 shows the coverage achieved as a function of the encounter threshold, τ , for a node speed of 2 metres per second (roughly that of a jogger), and node densities 0.5, 3.5 and 6.5 (these values are not intended to represent any particular application; they are chosen merely as illustration). The differing behaviours of the Random Waypoint and the Manhattan Grid mobility patterns are compared.

The figure quantifies the extent to which the coverage can be improved by increasing τ . At low densities, where flooding performs very poorly (coverage close to 0 when $\tau = 0$), the improvement is considerable; at high densities, flooding performs quite well and the gain of increasing τ is correspondingly smaller. The Manhattan Grid mobility pattern tends to produce lower coverage than Random Waypoint when the node density is low; the situation is reversed at higher densities.

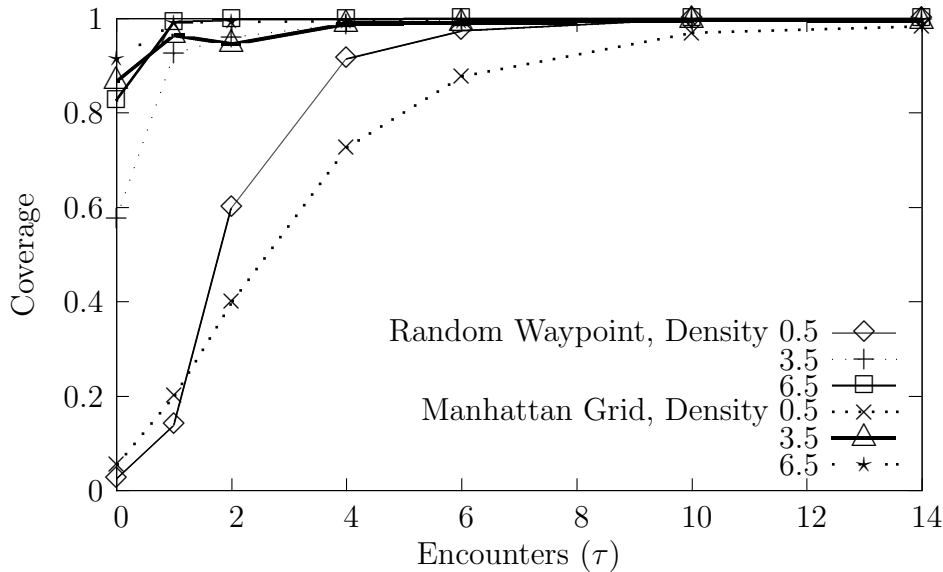


Fig. 1. Coverage vs Encounters

Other experiments (see [6]) have shown that higher node speeds (up to 100 ms^{-1}) have little or no effect on the coverage function.

Consider the analytical predictions concerning τ . For these 64 nodes, the encounter threshold given by equation (6) is $\tau = 10$, and the figure indicates that threshold does, indeed, achieve coverages close to 1. In fact, when the density is high, the threshold provided by equation (6) is rather conservative. This is because, for those densities, assumption (b) in Section 3 is too pessimistic.

In figure 2, the probability of achieving full coverage is plotted against the encounter threshold (the other parameters are the same as in figure 1).

That probability is estimated as the fraction of runs that were observed to achieve full coverage (remember that 25 runs are carried out for each point in the graph). In other words, express the result of the i th run as

$$p_i = \begin{cases} 1 & \text{if full coverage was achieved} \\ 0 & \text{otherwise} \end{cases} .$$

Then compute

$$P = \frac{1}{25} \sum_{i=1}^{25} p_i .$$

It should be pointed out that this performance measure is quite susceptible to random fluctuations. If just one of the 64 nodes fails to receive the message, the

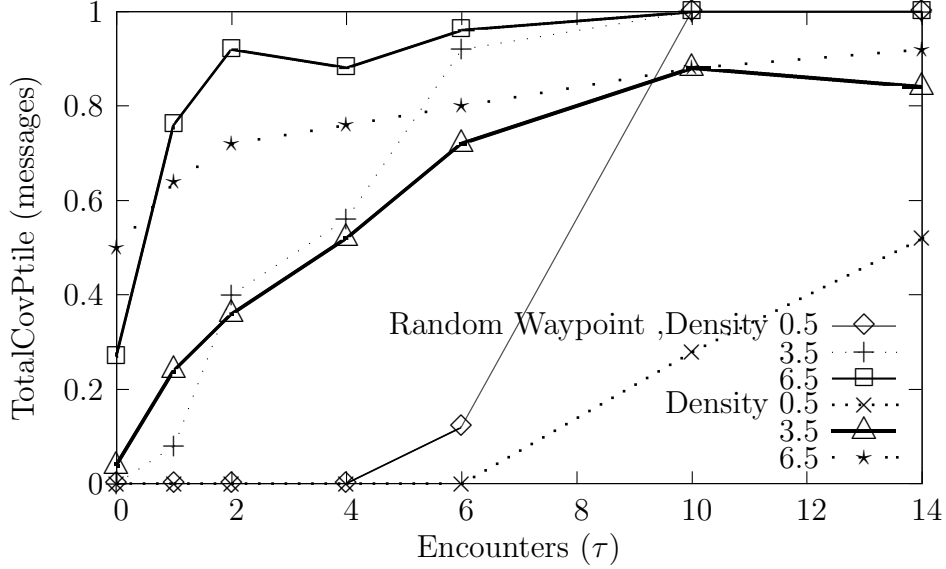


Fig. 2. Fraction of runs achieving full Coverage

corresponding p_i is set to 0 and P drops by $1/25$. Nevertheless, it is interesting to observe how different densities and mobility patterns affect this aspect of performance. The threshold level $\tau = 10$ suggested by the theory is sufficient to achieve full coverage with probability 1 under the Random Waypoint mobility pattern (even at the lowest density of 0.5). However, achieving full coverage for the Manhattan Grid appears to be much more difficult and requires considerably higher thresholds.

Other experiments have shown that, at any density, increasing the node speed tends to increase the probability of achieving full coverage.

The next performance measures to be examined, in the same experimental setting, are the average response time (time until algorithm terminates) and the average propagation time (time until maximum coverage is reached). Figure 3 shows these quantities plotted as functions of τ , in the case of the Random Waypoint mobility pattern.

An interesting aspect of the figure is that, while the response time keeps increasing with τ (as expected), the propagation time increases up to a point, and then decreases. To explain that behaviour, note that at low thresholds, the probability of full coverage is small and therefore the propagation time is equal to the response time. At higher thresholds, full coverage is achieved, and propagation completes, before nodes have stopped broadcasting. Moreover, further increases in τ tend to speed up the propagation, but prolong the response time. The figure also shows that the higher the node density, the earlier that divergence in timings occurs.

The situation is similar for the Manhattan Grid, as illustrated in figure 4.

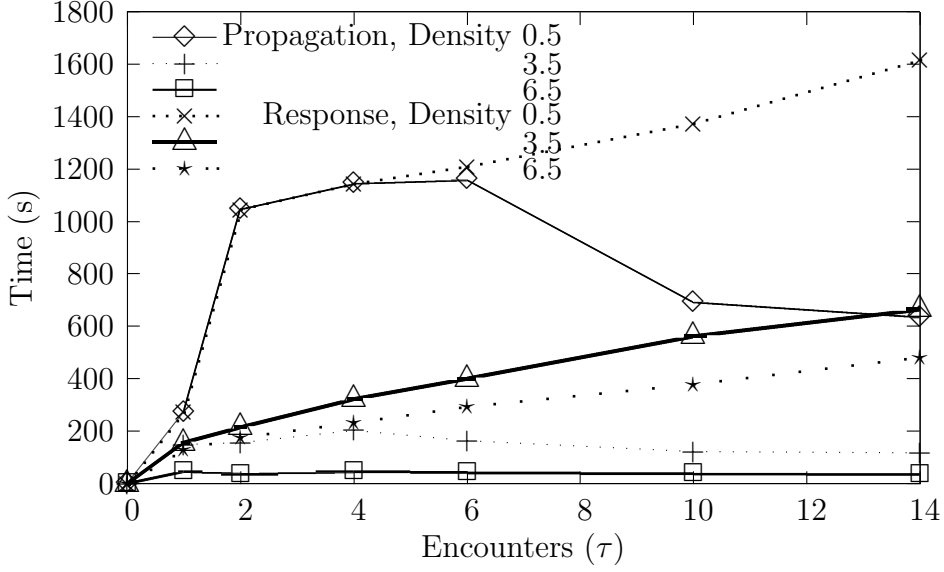


Fig. 3. Propagation and Response times: Random Waypoint

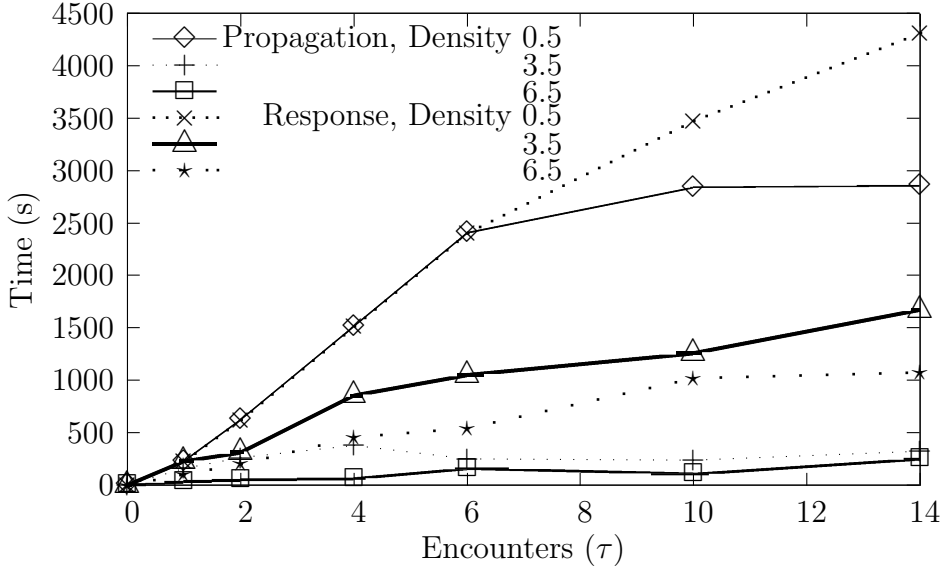


Fig. 4. Propagation and Response Times: Manhattan Grid

All timings are now larger, which is consistent with previous observations that message propagation is more difficult for the Manhattan Grid.

It can be expected that both the average response times and the average propagation times would decrease when the node speed increases. This has indeed been observed to be the case.

It is of some interest to follow *the process* of propagating a message among the nodes in a network. This is illustrated in figure 5, where the speed (60 ms^{-1}) and threshold ($\tau = 14$) are fixed, while the density is varied in the range 0.5 – 6.5. The graphs show how the rate of propagation changes as more and more

nodes are covered. At high densities, it takes longer to cover the last 5% of the nodes than the first 95%. This phenomenon is due to the fact that some nodes on the periphery of the terrain can be relatively more difficult to reach than the others. It is less pronounced at lower densities, but is still in evidence: the last 20% of the nodes take about as long to cover as the first 80%.

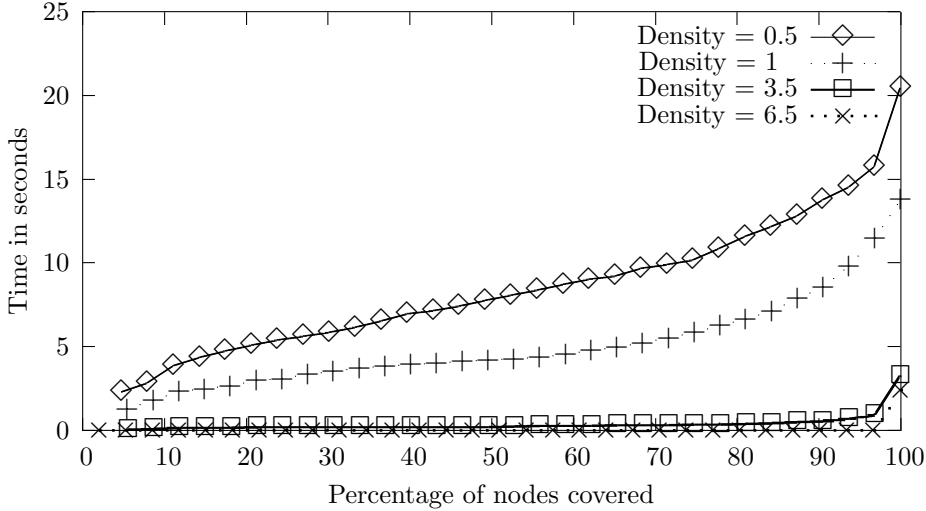


Fig. 5. Process of propagation

The next set of experiments aims to evaluate the savings in redundant broadcasts that can be achieved by the improvements suggested in section 4. The effect on coverage was also measured, but the results are not displayed because the changes were very small.

Figure 6 illustrates the effects of state-based improvement policies, under Random Waypoint mobility, for a low node density and a high one, and for the node speed of 2 ms^{-1} . The policies considered are: RAD, α -reduction and a joint application of both. The RAD period was distributed uniformly on the interval $(0,100) \text{ ms}$. The numbers of redundant broadcasts *per node* are plotted as functions of the encounter threshold.

We observe that at the low density of 0.5, the savings in redundant broadcasts achieved by the RAD and α -reduction policies are approximately 25% and 15% respectively. The joint application of RAD and α -reduction yields a small additional improvement.

On the other hand, when the density is at the high level of 6.5, the RAD policy becomes poorer than α -reduction policy as τ increases. When $\tau = 3$ (where all policies provide full coverage [7,8]), both RAD and α -reduction on their own achieve approximately 50% reduction in redundant broadcasts, while jointly they increase the saving to about 70%. A notable feature of figure 6 is that the average number of redundant broadcasts per node for the α -reduction policy,

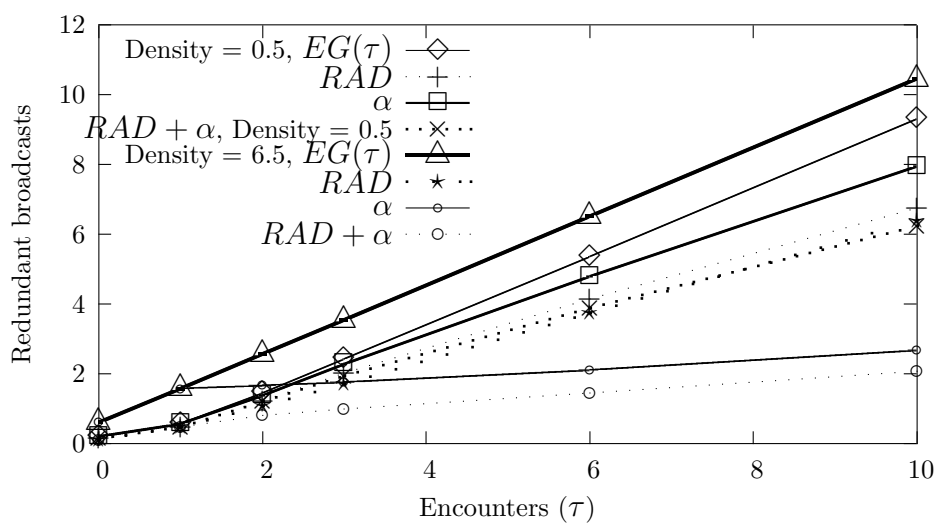


Fig. 6. Redundant Broadcasts: RAD and α -reduction policies; Random Waypoint; $2ms^{-1}$

and for RAD with α , is at high density almost independent of the value of τ . This feature is also observed to hold in experiments we carried out for different node speeds. For the speed of up to $40ms^{-1}$, figure 7 depicts the effects of these policies and figure 8 confirms that the reduction in redundant broadcasts is not achieved at the expense of coverage.

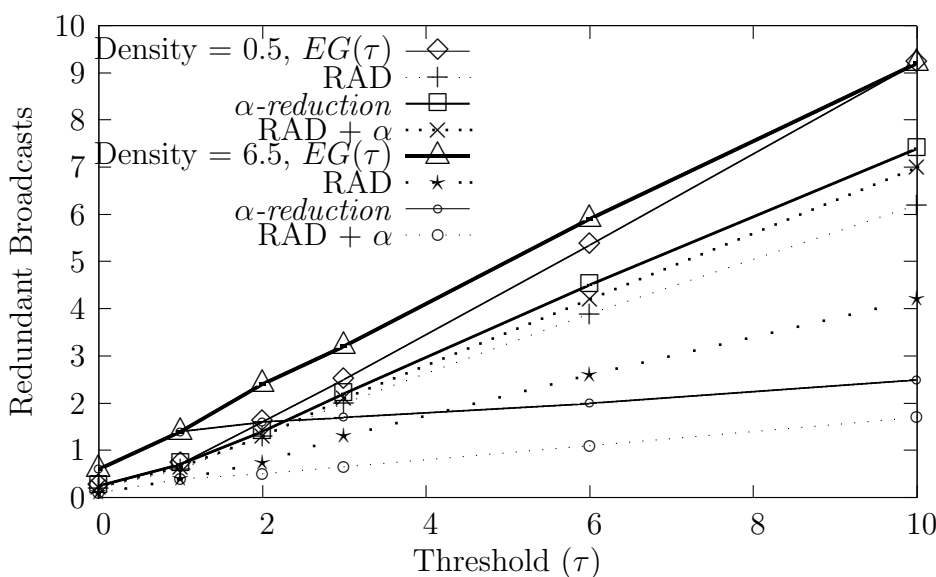


Fig. 7. Redundant broadcasts: RAD and α -reduction; Random Waypoint; $40ms^{-1}$

The other two suggested improvements suggested in section 4 were Encounter History (EH) and Propagation History (PH). Figure 9 compares the performance of these policies with that of the basic protocol (for speed $2ms^{-1}$). In this experiment, there is no limit on the number of node ids that can be

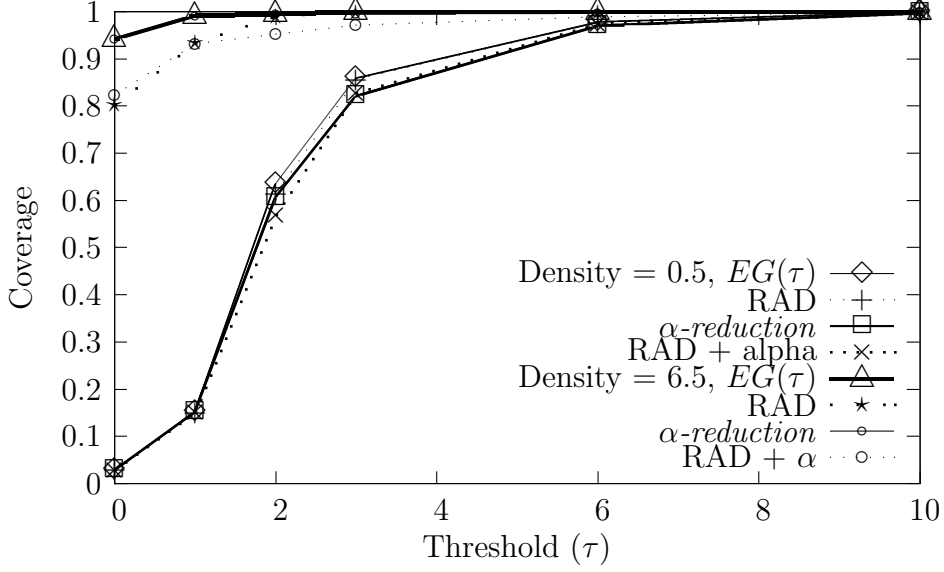


Fig. 8. Coverage: RAD and α -reduction; Random Waypoint; $40ms^{-1}$

attached to a message (i.e., $k = n$). Thus, any benefits in redundant broadcasts achieved by the PH policy should be set against the extra overhead of broadcasting longer messages than necessary.

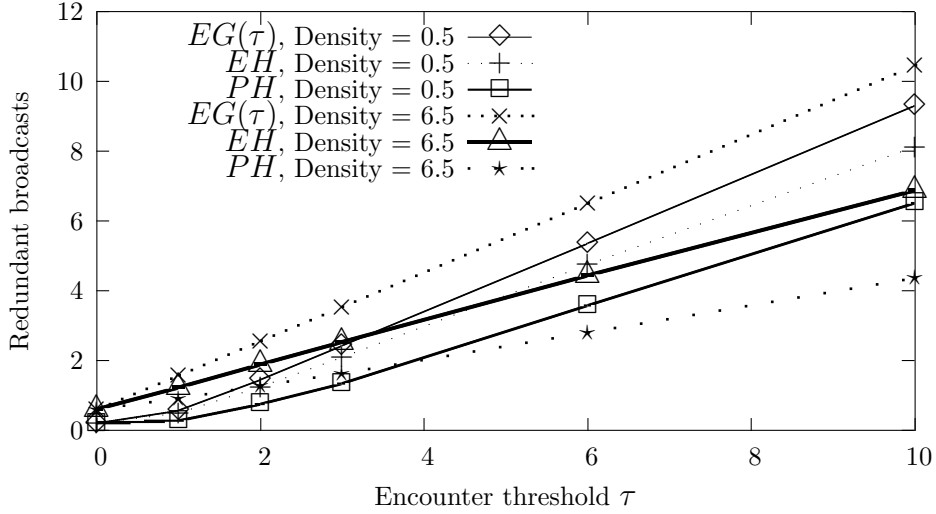


Fig. 9. Redundant broadcasts: History policies; Random Waypoint; $2ms^{-1}$

In this example, when the density is low, all three policies achieve full coverage at about the same threshold level, $\tau = 10$. There is some loss of coverage under the EH and PH protocols at lower thresholds, but it is not large. For $\tau = 10$ and higher, the EH policy reduces the average number of redundant broadcasts by just over 10%, while the PH policy reduces them by approximately 30%.

When the node density is high, full coverage is achieved for ($\tau = 2$ or $\tau = 3$).

Then the *EH* policy reduces the redundant broadcasts by about 50%, while the reduction achieved by the *PH* policy is close to 70% (but remember the comment about the extra overhead involved).

The above experiments were also carried out in the context of the Manhattan Grid mobility model. The outcomes are similar in character. The losses in coverage tend to be slightly larger, but the gains in reduced redundancy are quite a lot larger. We refer the reader to [7,8] for full details on the impact that mobility model had.

Attempting to reduce the number of redundant broadcasts used in the propagation of a message is clearly a worthwhile effort. The gains are perhaps not very important when the node density is high, since only a few broadcasts per node are then enough to achieve full coverage. However, when the density is low, one needs to set a large threshold on the number of broadcasts per node in order to achieve a satisfactory coverage. Under those conditions, when many of the broadcasts made are redundant, even a modest percentage reduction of the latter is significant in absolute terms.

Our experiments have shown that, at low node density, the average number of redundant broadcasts per node can be reduced by about 30% in the Random Waypoint mobility model, and by twice that amount in the Manhattan Grid model.

An interesting observation is that, at the threshold levels that are necessary to achieve high coverage, the simple policies — RAD and α -reduction — perform no worse than the ones employing caches and message lists (*EH* and *PH*). In fact, the α -reduction policy, with or without the addition of RAD, can be used with a high threshold over a range of densities and speeds, and still produce large savings in redundant broadcasts.

The next two experiments simulate the propagation of multiple messages in parallel, as discussed in section 5. Each run starts with 100 messages originating from each node, and terminates when no node can propagate the messages further. The node density is low, 0.5, which makes the task of propagation non-trivial. The aim is to compare the performance of the simple algorithm $SEG(\tau, M)$ with that of the modification $EG(\tau, M)$.

Figure 10 shows the coverage achieved by the $SEG(\tau, M)$ algorithm for different threshold values, and for different node speeds. Intuitively, the problem of redundant transmissions caused by new arrivals and premature departures during broadcasts should become more noticeable at higher speeds. That is indeed confirmed by the figure. Whereas a threshold of 4 is sufficient to achieve more than 99% coverage at the low speed of $2ms^{-1}$, one needs $\tau = 6$ for a similar coverage at $20ms^{-1}$ and $\tau \geq 9$ when the speed is $60ms^{-1}$ or more.

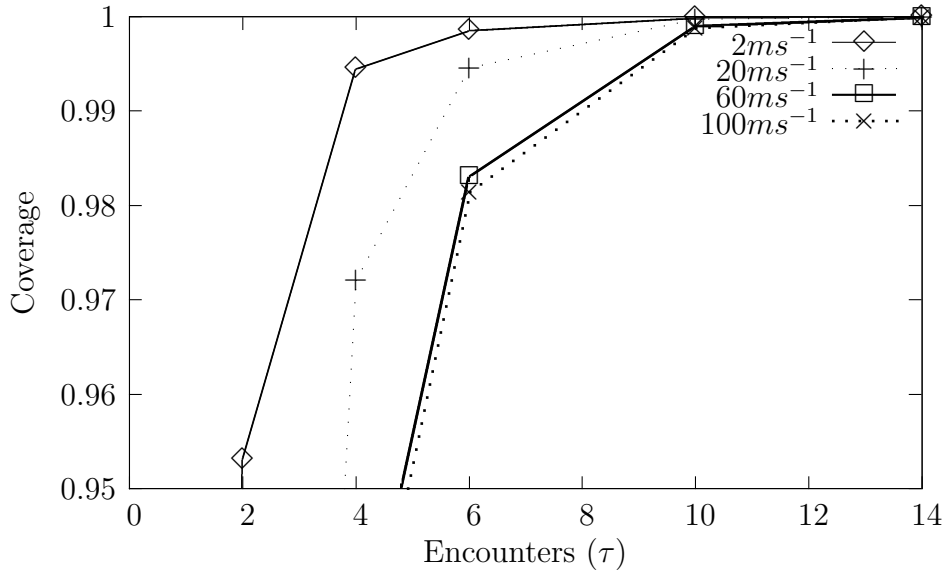


Fig. 10. Multiple messages: simple algorithm ($SEG(\tau, M)$); density = 0.5

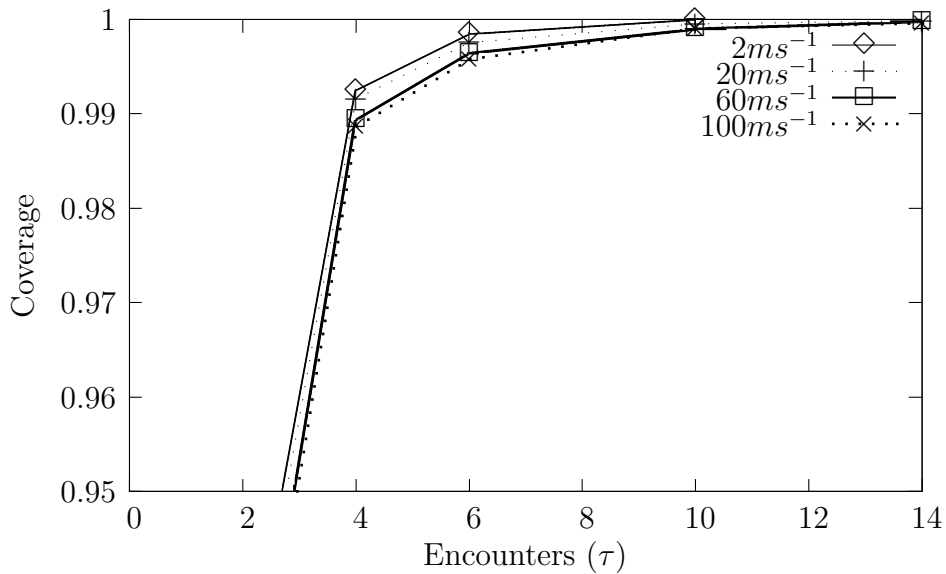


Fig. 11. Multiple messages: modified algorithm ($EG(\tau, M)$); density = 0.5

The comparative performance of the modified algorithm $EG(\tau, M)$ is illustrated in figure 11. The most notable feature of these results is that the coverage is now almost independent of the node speed; a threshold of 4 suffices to achieve about 99% coverage at all speeds. Thus, the performance penalties incurred by the simple algorithm at high speeds have been eliminated.

6.1 Comparison with Hypergossiping

In this section we compare the performance of Encounter Gossip with that of Hypergossiping [17], which is in many ways similar. On detecting an encounter, Hypergossiping establishes the need for message transmissions by first exchanging information on recently received messages. These exchanges are piggybacked onto successive hello beacons and are not counted in transmission cost. Hypergossiping thus attempts to save transmissions at the cost of delays before transmitting.

We use the experimental data reported in [17], figures 11 (a) and (b). We perform the same experiments as in that paper and compare the results on coverage achieved and transmissions carried out. Hypergossiping parameters not relevant to our protocol, such as message lifetime, are ignored. The Encounter threshold value τ is chosen as indicated in section 3.

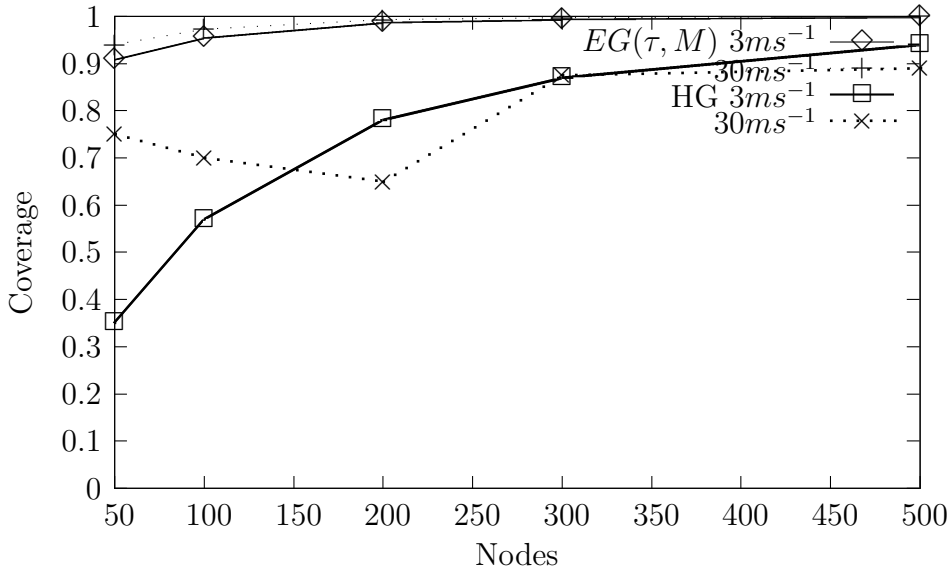


Fig. 12. Coverage vs Nodes

Figure 12 shows the coverage achieved by Encounter Gossip and Hypergossiping for two different node speeds, and different node densities. We observe that in all corresponding cases, Encounter Gossip provides significantly higher coverage.

The improved performance in terms of coverage is of course paid for by higher number of transmissions. This is illustrated in figure 13, which plots the average number of transmissions per node under the two protocols, for different node densities. At low densities, Encounter Gossip carries out 5-6 times more

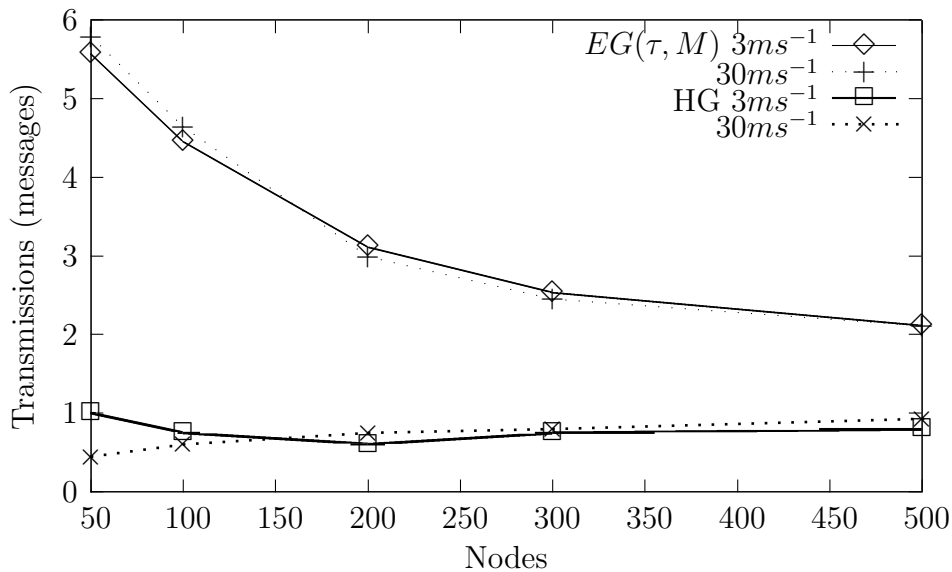


Fig. 13. Transmissions vs Nodes

transmissions than Hypergossiping, whereas that factor comes down to about 2 at high densities.

7 Experiments with a real-life MANET

Some of the assumptions we have made in this paper, and incorporated in the simulations described so far, may be difficult to defend in the context of a real-life ad-hoc network. Among the issues that have not been addressed here but may affect performance are:

Fading and transient network links [4]: Once a connection has been established between two nodes, even without mobility, the ability to transmit between them is not constant. Successful transmission of a packet over a wireless network link is probabilistic at best; having experienced an encounter, one cannot guarantee the quality or duration of the resulting link.

Communication grey zones [18]: A node that is able to receive a transmission from another node is not necessarily able to transmit a successful reply. The factors that contribute to the forming of communication grey zones include packet size (small packets are more likely to be transmitted successfully than large ones), fluctuating links (especially near the limit of the wireless range) and unequal ranges (e.g., node 1 is within the reach of node 2, but not vice versa).

Realistic mobility: The assumption of a flat world with either Random Waypoint or Manhattan Grid mobility pattern may be unduly simplistic. Humans rarely move at random; they tend to follow paths whose lengths and shapes are dictated by some underlying objectives, as well as by external events.

In an effort to establish whether the $EG(\tau, M)$ protocol would work in a real-life environment, we carried out an experiment where 21 users communicated using 20 PDAs and one laptop (all running the 802.11b protocol), while moving among rooms situated on three floors of an office tower block. In order to handle transient links and communication grey zones, it was decided to require the reception of h consecutive beacons before concluding that a node is a neighbour. Moreover, if no beacons are received from an established neighbour during a timeout interval of length T , that node is deemed to have left the neighbourhood. After some experimentation, those parameters were set to $h = 5$ and $T = 2.5$ seconds. The inter-beacon interval was 0.5 seconds.

The experiment was run twice, on consecutive work days to observe different mobility scenarios. On each day, users collected a PDA at about 9.30 am and dispersed about their usual daily routine. The devices were returned after 1 pm when their batteries had been depleted. PDA battery life was measured in earlier tests to range between 2.5 and 3 hours.

The model in section 3 suggests a threshold value of $\tau = 7$ for this number of users. Given their low density, low speed of movement and the obstacles to transmission inherent in the architecture of the building, that threshold was increased a little, to $\tau = 8$.

Each device was programmed to send 6 messages, at 10 minute intervals, following activation. This should allow about 2 hours to propagate the messages. In the event, 3 PDAs failed as their batteries were unable to power them for longer than half an hour. Also, 2 users on the first day and 3 on the second left the area and did not return during the experiment. They were excluded from the statistics.

The observed coverage on the first working day was 62%. This number represents the fraction of nodes that received a message, averaged over all sending nodes and all 6 messages they attempt to broadcast. The similar quantity for the second day was 59%.

A more detailed picture of what happened on the first working day is presented in figure 14. Each bar in the figure shows the average coverage achieved by the corresponding originating node. This is further partitioned into sections indicating the relative success of each message. Thus, if the height of a bar is 0.8 and the portions within it are of equal size, that means that the corresponding node achieved an average coverage of 80% and the coverage of each of its

messages was 80%.

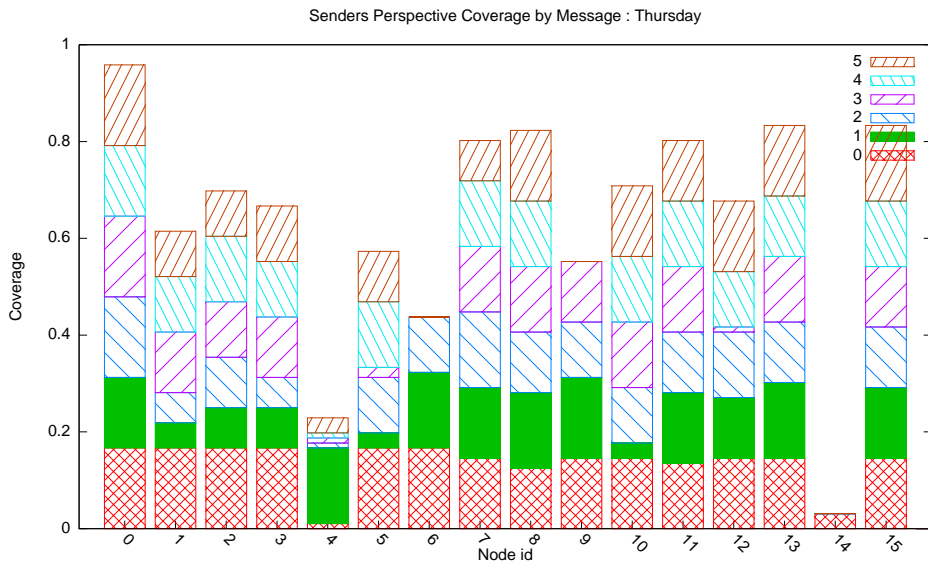


Fig. 14. Senders Coverage per message

Node 0, which was the user carrying the laptop, i.e. the most powerful transmitter, achieved a very high coverage for all its messages. The other nodes had varying degrees of success, with node 14 only propagating its first message, while node 4 performed poorly overall, despite achieving almost full coverage for its second message.

For comparison, figure 15 displays the coverage from the point of view of the receiving nodes. Now the height of each bar is the average fraction of sending nodes whose messages were successfully received by the corresponding receiver. Again, this is further subdivided into portions indicating the relative success of each message.

The picture revealed by this figure is quite different. Node 0 is now among the poor performers, managing to receive only about 40% of the messages sent, while nodes 4 and 14 are among the better performers, receiving 60% or more of the messages. Note, however, that whereas the set of messages concerning a given originator is of size 6, that concerning a given receiver is of size 90 (15 originators \times 6 messages each). The subdivisions of each bar refer to message *indices*, not individual messages.

We feel that, despite its small scale, this experiment has demonstrated the feasibility of encounter-based propagation in a real-life environment. The complete set of figures for this real world experiment may be found in [5]

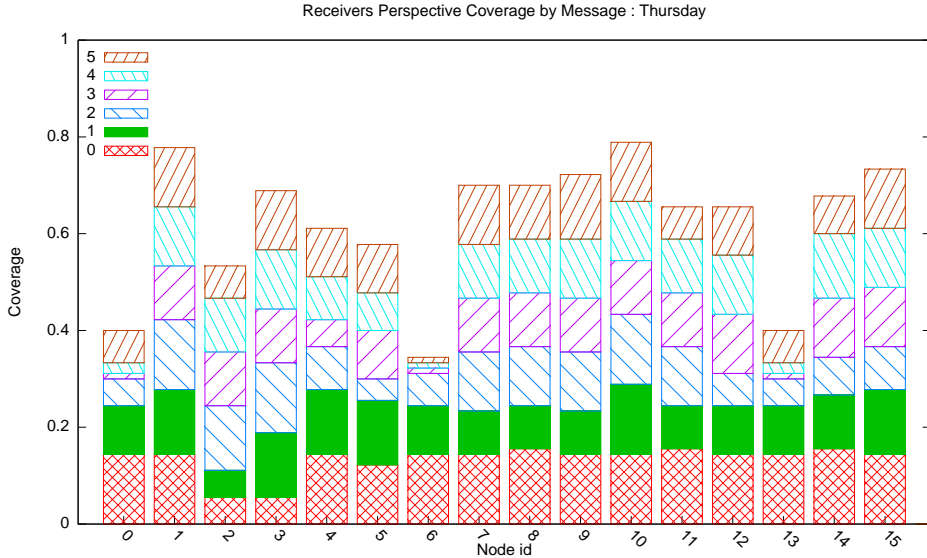


Fig. 15. Receivers Coverage per message

7.1 A Comparative Assessment

It is of interest to consider our experimental observations in the light of recent work done on the nature of human mobility and the implications for encounter-based (or opportunistic forwarding) protocols [2,15,1,19,13]. It has been shown [2,15] that when human mobility is purpose-driven, as opposed to random, the distribution of the inter encounter interval (X) - the interval between two successive encounters for a given node pair - approximately follows a *power law* which is characterised by its coefficient on log-log graphs. This is in contrast to many random mobility models found in the literature where the distribution of inter-encounter interval is regarded to be exponential [11]. (We also take this view in our analytical approximation for τ , in Section 3.)

The distribution of inter encounter intervals experienced by nodes in our experiments are shown in figures 16 and 17. In both experiments, we find that the distribution follows an approximate power law and the coefficient $cf = 2$ provides a much better fit.

Chaintreau et al [2] analyses the relation between cf and unicast latency (one-to-one transfer delay). They consider a generic 2-hop protocol in which the sender entrusts a subset of encountered nodes to relay its message to the destination, if the latter is encountered by the relay nodes. Four mobility traces are analysed: two from self run experiments, one from Dartmouth [13] and one from UC San Diego [19]. From their analysis, we can infer that $cf = 2$ is a large enough value for message propagation and that effective propagation of multiple messages is also influenced by *encounter duration* (D) which is the duration for which an encounter between a node pair lasts.

Figure 18 shows the distribution of encounter duration (D) measured in our experiments. (For example, in the Thursday experiment, about 30% of encounters lasted longer than 1 second.) The values of D we have observed are smaller than iMote experiments reported in [2] and are very much smaller than Dartmouth [13] and UCSD experiments [19] reported in the same paper. (The maximum encounter duration observed was **25** minutes in our experiments whereas 3 hours in iMote experiments.) The small values of encounter duration observed in our experiments can be attributed mainly to the unhelpful features of the multi-storeyed terrain where we ran our experiments. The tower block has reinforced concrete walls which cause significant barriers to wireless transmission. Therefore, opportunities for encounters arose mainly out of work-driven mobility and visiting common places such as Coffee rooms and Water Closets. Whereas, those reported in [2] were done in a modern, flat structure. Consequently, the coverage we observed were only 62% and 59% - not close to 1.

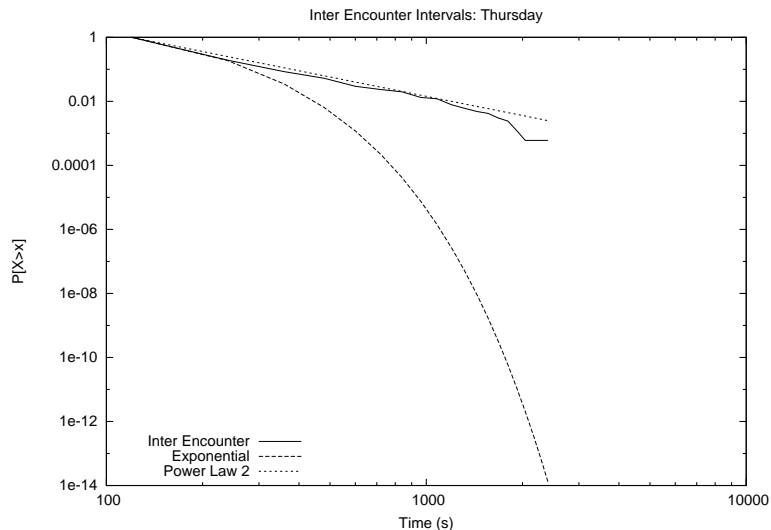


Fig. 16. Inter-encounter intervals (X): Thursday

8 Conclusions

The main contributions of this paper can be summarised as follows:

1. Introduction of the family of encounter propagation protocols (Section 2).
2. Mobility-independent estimate for the value of τ that achieves high coverage (Section 3, equation (6)).
3. Improvements to the protocol aimed at reducing the number of redundant transmissions (Section 4). In particular, a combination of RAD and α -reduction is both simple and efficient, and hence is worth implementing.
4. An efficient generalisation to multiple propagations in parallel (Section 5).

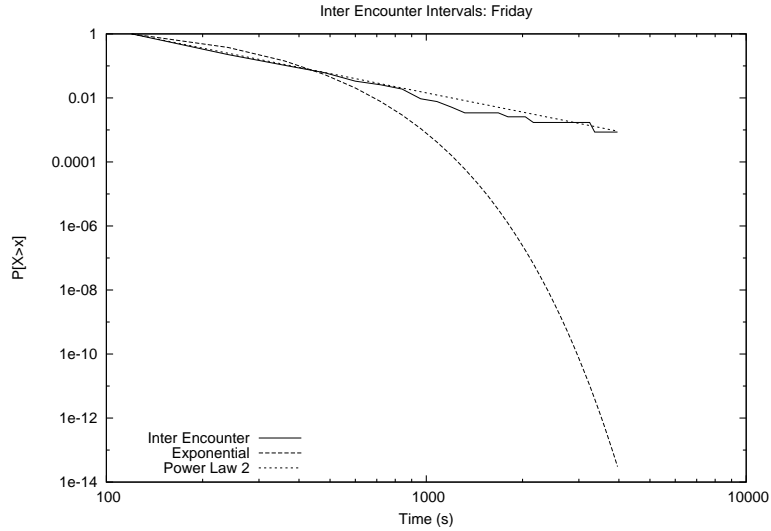


Fig. 17. Inter-encounter intervals (X): Friday

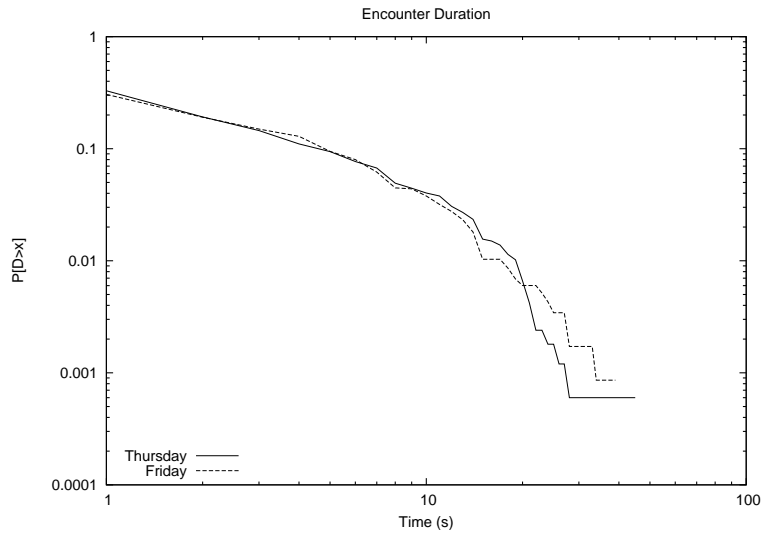


Fig. 18. Encounter Durations (D)

5. Quantitative performance results obtained by simulation (Section 6).
6. A limited validation of the protocol in a real-world environment (Section 7).

The differences observed between the Manhattan grid and random waypoint models illustrate the importance of the mobility pattern to the performance of the propagation protocols. It is clearly desirable to devise and experiment with patterns that would be realistic in different applications.

One could think of improvements to the protocol involving a FIFO buffer for multiple messages. For example, the number of times a message is broadcast by a node might change dynamically in response to changing conditions. That number could be adjusted by keeping track of repeated receptions of the same

message. A time-out interval can be introduced, to force the discarding of a message if the node does not experience a sufficient number of encounters. In addition, the encounter threshold may be controlled by the number of nodes already encountered, and possibly by the mobility pattern. All these are worthy topics for future research.

References

- [1] M. Balazinska and P. Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. *Proceedings of MobiSys 2003*, pages 303–316, 2003.
- [2] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. *University of Cambridge Computer Laboratory, Tech. Rep. UCAM-CL-TR-617, Feb, 2005*.
- [3] C. Chiang and M. Gerla. On-demand multicast in mobile wireless networks. *Proceedings of IEEE ICNP*, pages 260–270, 1998.
- [4] K.W. Chin, J. Judge, A. Williams, and R. Kermode. Implementation experience with MANET routing protocols. *ACM SIGCOMM Computer Communication Review*, 32(5):49–59, 2002.
- [5] D.E. Cooper, P. Ezhilchelvan, and Mitrani. A Real World Evaluation of Encounter Gossip. *Technical Report Series, CS-TR-1090*, 2008.
- [6] D.E. Cooper, P. Ezhilchelvan, and I. Mitrani. High coverage broadcasting for mobile ad-hoc networks. *The Third IFIPTC6 Networking Conference*, pages 100–111, 2004.
- [7] D.E. Cooper, P. Ezhilchelvan, I. Mitrani, and E. Vollset. Optimization of Encounter Gossip Broadcasting in Mobile Ad-Hoc Networks. *Technical Report Series, CS-TR-921*, 2005.
- [8] D.E. Cooper, P. Ezhilchelvan, I. Mitrani, and E. Vollset. Optimization of Encounter Gossip Propagation in Mobile Ad-Hoc Networks. *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 529–532, 2005.
- [9] JJ Garcia-Luna-Aceves and E.L. Madruga. A Multicast Routing Protocol for Ad-Hoc Networks. *Proceedings of IEEE INFOCOM*, 99:784–792, 1999.
- [10] T. Gopalsamy, M. Singhal, D. Panda, and P. Sadayappan. A reliable multicast algorithm for mobile ad hoc networks. *Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002.*, pages 563–570, 2002.
- [11] M. Grossglauser and D.N.C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 10(4):477–486, 2002.

- [12] J. Hahner, C. Becker, and K. Rothermel. A protocol for data dissemination in frequently partitioned mobile ad hoc networks. *Proceedings of IEEE ISCC 2003, Eighth International Symposium on Computers and Communication.*, pages 633–640, 2003.
- [13] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 187–201, 2004.
- [14] Christopher Ho, Katia Obraczka, Gene Tsudik, and Kumar Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 64–71, Seattle, WA, 1999.
- [15] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 244–251, 2005.
- [16] A. Khelil. A Generalized Broadcasting Technique for Mobile Ad Hoc Networks. *PhD Thesis, University of Stuttgart, Department of Computer Science, Distributed Systems Group*, 2007.
- [17] A. Khelil, P.J. Marrón, C. Becker, and K. Rothermel. Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. *Ad Hoc Networks*, 5(5):531–546, 2007.
- [18] H. Lundgren, E. Nordströ, and C. Tschudin. Coping with communication gray zones in IEEE 802.11 b based ad hoc networks. *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, pages 49–55, 2002.
- [19] M. McNett and G.M. Voelker. Access and mobility of wireless PDA users. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(2):40–55, 2005.
- [20] I. Mitrani. *Probabilistic Modelling*. Cambridge University Press, 1998.
- [21] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, 1999.
- [22] K. Obraczka, G. Tsudik, and K. Viswanath. Pushing the limits of multicast in Ad hoc networks. *21 st IEEE International Conference on Distributed Computing Systems*, 1:719–722, 2001.
- [23] E. Pagani and G.P. Rossi. Reliable broadcast in mobile multihop packet networks. *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 34–42, 1997.
- [24] M. Pandey and D. Zappala. The Effects of Mobility on Multicast Routing in Ad Hoc Networks. 2004.
- [25] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast Core Extraction Distributed Ad-hoc Networking. *Proceedings of IEEE WCNC 1999, Wireless Communications and Networking Conference.*, pages 1313–1317, 1999.

- [26] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks, April 2000.
- [27] K. Viswanath and K. Obraczka. An adaptive approach to group communications in multi hop ad hoc networks. *Proceedings of IEEE ISCC 2002, Seventh International Symposium on Computers and Communications.*, pages 559–566, 2002.
- [28] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.



David Edward Cooper received his BSc(hons) Computing Science in 2001 and his MSc System Design and Internet Applications in 2002 both at Newcastle University, UK. Continuing at Newcastle he received his PhD in December 2008 and is currently a Research Associate of the University. His current research interests include Mobile Ad Hoc Networks and Dynamic Deployment of Web Services.



Paul Devadoss Ezhilchelvan holds a Personal Readership in Distributed Computing in the School of Computing Science, Newcastle University, from where he also received his Ph.D. in Computing Science in 1989. His research interests are primarily in the areas of dependability and fault-tolerance, distributed algorithms, group communication protocols, fair-exchange protocols, distributed agreement, simulations and performance evaluation. His research investigations cover both wired networking systems and mobile wireless systems.

He was the principal architect of VOLTAN fault-tolerant systems and NewTOP group communication systems.



Isi Mitrani studied Mathematics at the Universities of Sofia and Moscow, and Operations Research at the Technion, Haifa. He has spent most of his academic life at the University of Newcastle, where he is now an Emeritus Professor. His research interests are in the areas of probabilistic modelling, performance evaluation and optimisation. Publications include 7 books, more than 100 journal and conference papers and one patent.