

# ELIP: Embedded Location Information Protocol

Farid Benbadis, Marcelo Dias de Amorim, and Serge Fdida

Laboratoire LIP6/CNRS  
Université Pierre et Marie Curie  
8, rue du Capitaine Scott – 75015 – Paris – France  
{benbadis, amorim, sf}@rp.lip6.fr

**Abstract.** It has been recently shown that mobility in ad hoc networks can be an advantage instead of an inconvenience. Nevertheless, one class of mobile elements has been neglected up-to-date: *data packets*. In this paper, we propose to take advantage of the inherent mobility of data packets to disseminate location information throughout the network. We focus on the age-and-position based (APB) routing case. Knowing its own geographic or virtual coordinates is not enough since a source needs to discover the position of the destination before establishing a communication. This is the role of a location service, which depends, in turn, on an efficient location distribution/publishing system. Our proposal, Embedded Location Information Protocol (ELIP), allows nodes to piggyback their coordinates in existing data packets in order to efficiently disseminate their positions in the network. Contrary to traditional approaches that depend on encounters between nodes, ELIP converges much faster and does not require permanent node mobility.

## 1 Introduction

In large scale mobile ad hoc networks, position-based routing has proven to be efficient because of its simple forwarding policies. Indeed, nodes make elementary forwarding decisions based solely on the coordinates of their direct neighbors and of the destination. This avoids the need for topology knowledge beyond one-hop. Since there is no need for maintaining explicit routes, this type of routing algorithm is scalable and robust to mobility.

Position-based routing algorithms are composed of three main steps. The first one is *positioning*, where each node determines its coordinates using an absolute positioning system like GPS (Global Positioning System) or a GPS-free relative positioning algorithm [1, 2, 3]. The second step is the *location service*, used by a source to obtain the destination's coordinates. It is important to note here that the location service is composed of two distinct operations (cf. Section 2): dissemination of location information and lookup. The third step is *forwarding*, where next-hop decisions are based on the destination's position and on the position of the forwarding node's neighbors. Examples are the Compass routing [4], Restricted Directional Area, used in [5], or Greedy Perimeter Stateless Routing (GPSR) [6]. Refer to [7] for a comprehensive survey on such algorithms.

In this paper, we focus on the *dissemination* model associated with the location service. The dissemination model dictates the degree of location information replication throughout the network. This replication may range from null, meaning that a node is the only one to know its position, to full, where all nodes know the entire topology. Both cases have pros and cons. On the one side, zero replication does not require updates of information that are never used, but results in a lookup procedure that generates a great amount of control traffic overhead. On the other side, total replication results in zero overhead for lookups but high overhead for maintenance, especially in networks of mobile nodes.

A more recent way of estimating locations in ad hoc networks without incurring much traffic overhead is to use age-and-position based (APB) algorithms. In such algorithms, every node maintains a local database where it records the identifier and location of other nodes in the topology. Each location is associated with an *age*, which gives the time elapsed since the last time the location information has been updated. The local database is consulted to obtain approximate coordinates of the destination's current position. In such an approach, a node sends packets to the destination's position it knows. These packets are rerouted by nodes that have fresher location information until they are received by the destination [8]. Contrary to traditional approaches where the data transmission phase comes after the location phase, in APB methods the destination's position discovery is achieved during packet forwarding. It is clear that the lower the age, the better the estimation of the node's location. The problem in APBs is then to find an efficient way of distributing good estimations of node positions in the network.

In [8], Grossglauser and Vetterli propose to use encounters as a way of disseminating location information. In such an approach, nodes update their local databases each time they are directly connected to other nodes. The advantage of using encounters is that it results in near-zero location dissemination overhead. Nevertheless, as we will see in Sections 2 and 4, it results in high overhead in the lookup phase. Furthermore, the efficiency of this approach is closely related to the mobility model of the nodes.

In this paper, we do not propose a novel APB routing protocol, but an efficient dissemination mechanism of nodes' coordinates that can be used in any APB protocol. This algorithm is *Embedded Location Information Protocol* (ELIP). It uses *existing* data packets to disseminate location information. ELIP relies on the basic assumption that packets are much more mobile than nodes. ELIP piggybacks nodes coordinates with existing data packets, incurring little traffic overhead to better disseminate topology knowledge than encounter methods. We will show that, although ELIP incurs little overhead for disseminating location information, it largely reduces the global overhead (dissemination+lookup) when compared to the encounter approach.

Our algorithm is efficient for these three reasons: (1) a node communicates with several other nodes. Thus, it is likely that it sends packets containing its coordinates in different directions, which results in packets traveling across several other nodes in the topology; (2) nodes move and forward packets in different

regions of the topology, which leads to a wider dissemination pattern; (3) the overhead generated by the headers insertion is negligible when compared to the reduction of overhead signaling messages during the lookup phase.

Our results show that using ELIP, instead of encounters, leads to much lower average ages of location information in the network. The consequence is that the lookup phase generates lower discovery traffic overhead and delay. Furthermore, we observe that the resulting path lengths are about 30% shorter with ELIP. We also show through a number of simulations that the global overhead in ELIP is upper bounded by the encounter-based approach.

The remainder of this paper is organized as follows. In Section 2, we present the context where the ELIP algorithm can be used. Section 3 details the main components of ELIP and how nodes use existing data packets to disseminate location information. In Section 4, we evaluate the performance of our proposal with different topology densities and mobility models, and compare it with the encounter-based approach. Finally, Section 5 concludes the paper and provides some ideas for future work.

## 2 Location = Dissemination + Lookup

A location service is composed of two main components: dissemination and lookup. The dissemination model refers to the ability of the location service to distribute location information throughout the network. ELIP falls into this category. The lookup phase consists in obtaining the location information of a destination from a node that has been provided with this information during the dissemination phase. In the following we will present different ways of implementing a dissemination model as well as the lookup algorithm that we will use to evaluate ELIP.

### 2.1 Disseminating location information

As stated above, the efficiency of the location service depends on an adequate system to disseminate location information in the network. This task can be performed in different ways according to how many nodes play the role of a location server. In *all-one* approaches, every node knows only its own position. This method requires that the source floods a route request in the network until the destination responds with its coordinates. This is clearly not scalable because of the high traffic control overhead generated. Examples of protocols that use such an approach are DSR [9] and AODV [10]. *Some-some* methods distribute the whole topology information among a subset of the nodes. When a node looks for a destination, it sends a route request to one of the servers. Although simple, this approach is considered unfair in the ad hoc concept because some nodes have more responsibilities than others [7]. In the *all-some* algorithms, every node in the topology plays the role of a rendezvous point. This category includes DHT-based location services, where a node  $n$  stores its location information in

a rendezvous node  $r$  depending on  $n$ 's identifier [11, 12].<sup>1</sup> The problem in such a system is that, in order to keep an accurate location system, the location information must be updated every time a node moves. Depending on the dynamic nature of the topology, this may lead to high signaling overhead. Finally, there is the *all-all* approach, where each node always knows the positions of all the other nodes in the topology in a proactive fashion [13]. While discovering a node's position is fast and does not generate any traffic overhead, such a solution leads to high traffic overhead in order to keep databases constantly updated.

## 2.2 Age- and position-based (APB) routing

In APB routing protocols, we essentially use the age of location information to compute routes from source to destination. Because nodes move, their positions change with time. Location information of node  $m$  stored in  $n$  must be then frequently updated. The older the information, the worse the location estimation.

The Last Encounter Routing (LER) has been proposed by Grossglauser and Vetterli as an implementation of the APB algorithm [8]. In LER, nodes do not exchange any explicit location information. The only available information a node has is the history of its encounters with other nodes and the ages of these encounters. They assume w.l.g. that two nodes have encountered each other if they have been directly connected in the past. In their proposal, the authors implement LER by using Last Encounters (LE) as the dissemination algorithm and Exponential Age SEarch (EASE) as the lookup method.

Our proposal, ELIP, is an alternative to LE. Thus, in order to fairly compare these two approaches, we use EASE as the lookup algorithm. We briefly describe EASE in the following. For further details, please refer to [8].

For two arbitrary nodes  $i$  and  $j$ , we note  $\tau_{i,j}$  the time elapsed since the last time  $i$  and  $j$  were directly connected, with the convention that  $\tau_{i,j} = \infty$  if  $i$  and  $j$  have never met, and  $\tau_{i,j} = 0$  if they are currently connected. We also note  $l_{i,j}$  the position where  $i$  has met  $j$  for the last time.

Consider a node  $s$  that wants to communicate with  $d$ . If  $\tau_{s,d} \neq \infty$ , then  $s$  sends the message to  $l_{s,d}$ , *i.e.* the  $d$ 's coordinates stored by  $s$  in its position table.<sup>2</sup> The node the geographically closest to  $l_{s,d}$ , called *anchor node*, will be responsible for determining the following coordinates the message must be sent to (*i.e.* the following anchor node). The idea is to make messages jump between anchor nodes with decreasing ages until the message is received by the destination.

Let  $\mathbf{A} = \{a_1, a_2, a_3, \dots\}$  be the set of anchor nodes for a message traveling from  $s$  to  $d$ . Let also  $\tau_{a_i}^\ominus$  be the age indicated in the message received by anchor node  $a_i$  and  $\tau_{a_i}^\oplus$  be the age of the new location information that anchor node  $a_i$  must compute. Clearly, we must guarantee that  $\tau_{a_i}^\oplus < \tau_{a_i}^\ominus$ .

The first test node  $a_i$  makes is to look into its own position table to verify if  $\tau_{a_i,d} < \tau_{a_i}^\ominus$ . If so, then  $a_i$  does  $\tau_{a_i}^\oplus := \tau_{a_i,d}$  and sends the packet to  $l_{a_i,d}$ .

<sup>1</sup> DHT = Distributed Hash Table

<sup>2</sup> We assume that the message is routed from  $l_s$  to  $l_{s,d}$  via a geographic forwarding protocol. We do not focus on this point in this paper.

Otherwise, it locally floods a  $d$ 's *position request* with some TTL (time-to-live). Node  $a_i$  receives a response only if at least one of the nodes within this scope has encountered  $d$  more recently than  $\tau_{a_i}^\ominus$ . If  $a_i$  does not receive any response, it increases the TTL and floods a second request. This procedure is repeated until node  $n_i$ , a neighbor of  $a_i$ , responds with  $l_{n_i,d}$ , *i.e.* the location of its last encounter with  $d$ , and  $\tau_{n_i,d}$ . Node  $a_i$  does  $\tau_{a_i}^\oplus := \tau_{n_i,d}$  and forwards the message to  $l_{n_i,d}$ , which will be received by the next anchor node  $a_{i+1}$ . The same search procedure is performed by  $a_{i+1}$ , and so on until the destination  $d$  is reached. It has been shown in [8] that the destination is reached after a number of steps of exponentially decreasing distances for a network with random node mobility.

### 3 Embedded Location Information Protocol: Algorithm details

Recall that the ELIP algorithm is designed to be integrated in an APB routing protocol for ad hoc environments where nodes are supposed to know their geographic (or virtual) coordinates. The goal of ELIP is to increase the dissemination degree of node location information, making it easier for a source to locate a destination. The originality of ELIP is its ability to widely disseminate location information and reduce the global overhead by piggybacking position and age information of mobile nodes with existing data packets without creating any new signaling packets.

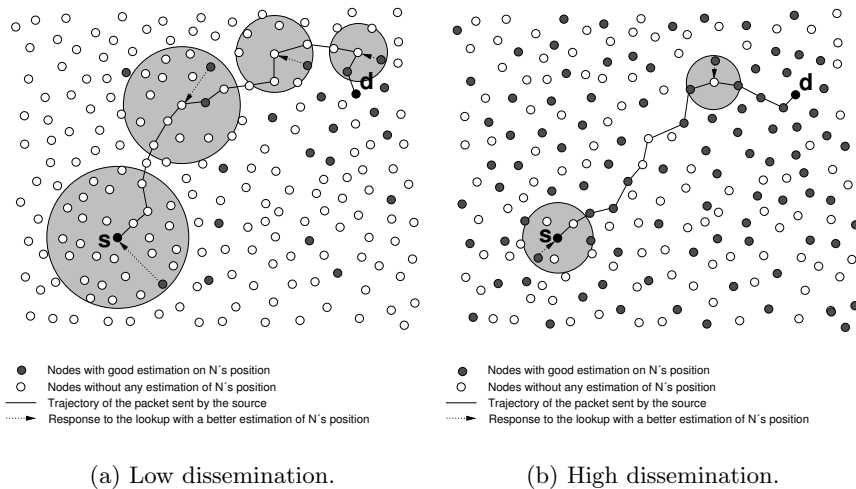
#### 3.1 Node mobility $\times$ Packet mobility

LER, contrary to traditional routing algorithms that have problems to deal with node mobility, takes advantage from it. The problem of LER is that its performance depends on the mobility pattern of the nodes. If nodes present low mobility or limited occupancy area, they are unable to encounter other nodes, but only a subset of the nodes in the same physical scope.

In ELIP, location dissemination is implemented by encounters between nodes and by including location information in data packets. Because a node communicates with several other nodes, data packets cover a larger area through different routes. All nodes in these routes are then able to update their routing tables even if nodes do not move at all.

Fig. 1 shows an example of the functioning of EASE in two different scenarios. In Fig. 1(a), the distribution of location information about the destination is low. Note that anchor nodes perform searches in relatively large zones, which incurs high signaling traffic overhead. In the second scenario, location information about the destination is well disseminated in the network. This clearly reduces the overhead generated during the lookup phase.

The good performance shown in the example of Fig. 1(b) can be achieved if nodes use ELIP because updated location information is carried by packets, instead of nodes. This is however difficult to obtain when using LE because nodes must permanently travel the entire topology in order to maintain a good



**Fig. 1.** Impact of the dissemination model on the lookup overhead.

distribution of location information throughout the network. The gains obtained by ELIP include: smaller latency for location discovery, limited overhead, and shorter end-to-end paths.

### 3.2 Algorithm

In ELIP, each node in the path between the source and the destination can perform two operations: read and write.

- Read: A node participating in the forwarding procedure consults the ELIP field in the packets and updates the corresponding entry in its position table.
- Write: Any node participating in the forwarding procedure is a candidate node to include its location coordinates in packets. This depends on the insertion model (cf. section 3.2.4) adopted by the nodes.

Let  $l_i = (x_i, y_i)$  be the geographic coordinates of node  $i$  and  $\mathbf{P}_i$  be the position table that  $i$  uses to store positions and ages about other nodes (we will explain in details in Section 3.2.3 how  $i$  obtains these positions). An example of  $\mathbf{P}_i$  is shown in Fig. 2. In this position table, the information node  $i$  has about  $j$  is  $A_{i,j} = [\text{ID}_j, \hat{l}_{i,j}, \tau_{i,j}]$ , where  $\hat{l}_{i,j}$  is a local approximation of  $l_j$  and  $\tau_{i,j}$  is the time elapsed since the last time  $i$  updated this information (*i.e.*, the age of  $\hat{l}_{i,j}$ ).

Existing data packets are encapsulated in ELIP packets with the structure shown in Fig. 3. In this packet,  $d$  is the destination node,  $l_d$  is an estimation of  $d$ 's position,  $\tau_d$  is the age of this estimation, and  $\Gamma$  is the current location of some node in the path traversed by the packet. The node that fills  $\Gamma$  depends on the

Node ID: $i$		
Dest. ID	$\hat{l}_{i,\cdot}$	$\tau_{i,\cdot}$
1	$(\hat{x}_{i,1}, \hat{y}_{i,1})$	$\tau_{i,1}$
2	$(?, ?)$	$\infty$
$\vdots$	$\vdots$	$\vdots$
$i$	$(x_i, y_i)$	0
$\vdots$	$\vdots$	$\vdots$
$j$	$(\hat{x}_{i,j}, \hat{y}_{i,j})$	$\tau_{i,j}$
$\vdots$	$\vdots$	$\vdots$
$N$	$(?, ?)$	$\infty$

**Fig. 2.** Example of a position table. For the sake of clarity, “(?, ?)” and “ $\infty$ ” indicate that node  $i$  has no estimation of the corresponding node.

insertion criteria presented in Section 3.2.4. Fields  $l_d$  and  $\tau_d$  are required by the lookup algorithm, which is common to both ELIP and LE. The only overhead introduced by ELIP is the  $\Gamma$  field, which contains the ID and the location of the node that has written in the packet. To prove that ELIP is more efficient than LE, we have to show that the overhead introduced by  $\Gamma$  is compensated by a smaller overhead generated during the lookup phase (cf. Section 4).

$l_d$	$\tau_d$	$\Gamma$	Original Data Packet
-------	----------	----------	----------------------

**Fig. 3.** Packet structure.

**3.2.1 Packet creation and forwarding** We assume in a first time that source  $s$  has an estimation of  $d$ 's position ( $d$  is the destination). After  $s$  creates the original data packet, it fills the  $l_d$  field with  $\hat{l}_{s,d}$  and the field  $\tau_d$  with  $\tau_{s,d}$ . The  $\Gamma$  field is filled or not depending on the criteria presented in the next paragraph. The packet is then ready to be sent using a geographic forwarding protocol. In the case where  $s$  has no location estimation of  $d$ , we have to implement a lookup algorithm. In this paper, we simply use the EASE algorithm as the lookup mechanism [8].

**3.2.2 Writing coordinates in a packet** Let  $P_{s,d} = \{s, p_{s,d}^1, p_{s,d}^2, \dots, p_{s,d}^k, d\}$  be the path between  $s$  and  $d$ , where  $k$  is the number of nodes in the path (excepting  $s$  and  $d$ ),  $s = p_{s,d}^0$ , and  $d = p_{s,d}^{k+1}$ . Node  $p_{s,d}^i$  has the possibility to

disseminate its location information,  $\Gamma_{p_{s,d}^i}$ , to every node  $p_{s,d}^j$ ,  $i < j \leq k + 1$ , by writing it in the  $\Gamma$  field of the data packets. In this paper, we assume that  $\Gamma$  can contain coordinates of only one node and that it is read-only, *i.e.* once a node has written its coordinates, no other node in the path can change this information. ELIP is also designed such that only a small fraction of data packets are filled with  $\Gamma$ , in order to keep the dissemination overhead small. The probability for a node of writing in a packet will be described in Section 3.2.4.

**3.2.3 Updating position tables** In ELIP, a node may update its position table in three different cases. First, it uses simple encounters as in the LE scheme. Each node stores the position of the last encounter with any other nodes. Second, nodes update their position tables using information carried in the  $\Gamma$  field of transit packets. Destination  $d$  as well as all nodes  $p_{s,d}^j$ ,  $i < j \leq k$ , update the  $\{\text{ID} = p_{s,d}^i\}$  entry in their position table by reading the information written by  $p_{s,d}^i$  in  $\Gamma$ . Third, any node in the path  $P_{s,d}$  can update location information about the destination if the header of a data packet in transit contains fresher information than the one in its local position table.

**3.2.4 Insertion probability.** Let  $\pi$  be the probability that a node write in a packet. In this paper, we consider that  $\pi$  is fixed and is the same for all the nodes. Let us consider a packet to be routed from source node  $s$  to destination node  $d$  through the path  $P_{s,d} = \{s, p_{s,d}^1, p_{s,d}^2, \dots, p_{s,d}^k, d\}$ . Since we apply a read-only policy, the probability for a node  $p^i$  to write its coordinates in the  $\mathcal{T}$  field is then  $\pi(1 - \pi)^i$ .

## 4 Analysis

We present in this section some simulation analysis of the proposed system.

### 4.1 Simulator model

We have conceived a network simulator to evaluate the efficiency of ELIP and LE in disseminating nodes' coordinates. We describe in the following the network model used in our simulations.

- **Topology.** The emulated network environment is a square universe of 1000 meters on a side, partitioned into a grid with squares of one square meter. Vertices of the grid are the positions where nodes can be placed. Nodes' initial positions are randomly chosen.
- **Neighborhood.** Each node in the network has as immediate neighbors all nodes in a range of  $r$  meters. We suppose links to be symmetric. An important parameter in our simulations is the relative density, given by the number of neighbors per coverage zone.



- **Mobility.** We use in our simulations the popular and commonly used random waypoint (RWP) mobility model. The waypoints are uniformly distributed in the area.
- **Time scale.** We assume that packets travel much faster than nodes. The topology is supposed then to be frozen during packet transfer.
- **Traffic quantity.** We have simulated both LE and ELIP for different values of traffic patterns, which is represented by the maximum number of pairs (source, destination) that communicate in parallel. This value is set as a percentage of the total number of nodes in the network. For instance, for 1000-node topology and a 10% traffic probability, the maximum number of simultaneous pairs (source, destination) is 100.
- **Forwarding algorithm.** We assume that nodes know their current coordinates and the ones of their direct neighbors. For forwarding, we use a classical geographic method. Suppose that node  $i$ , located in  $l_i$ , receives a packet to be forwarded to  $d$ , with  $l_d$ . Let  $N_i = \{n_1, n_2, \dots, n_l\}$  be the set of  $i$ 's neighbors and  $\delta(i, j)$  be the Euclidean distance between  $i$  and  $j$ . Node  $i$  forwards the packet to neighbor  $n_x$  located in  $l_{n_x}$  if  $\delta(n_x, d) < \delta(i, d)$  and  $\delta(n_x, d) \leq \delta(n_y, d)$ ,  $\forall n_y \in N_i$ .

## 4.2 Measurements

The simulator is written in REACTIVEML [14], a language dedicated to the simulation of complex dynamic systems.<sup>3</sup> The advantages of using REACTIVEML are twofold: efficient runtime and compact code. In the simulator, each node is a process that moves, discovers the neighborhood, and routes packets.

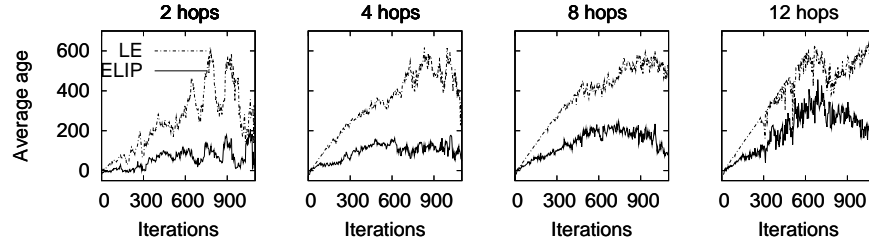
We have conducted a set of simulations in order to evaluate the efficiency of disseminating node locations with ELIP. Recall that we do not focus here on the performance of APB routing protocols, but on the advantages of using ELIP over the Last Encounter algorithm. In our simulator, each node has two position tables, one for LE and another for ELIP, and they use EASE as the lookup mechanism. This guarantees that both ELIP and LE are evaluated under identical conditions.

For each route, we measure: average age of location information, search depth, cumulative overhead, and path length.

**4.2.1 Average age of location information** We consider in the beginning of our simulations that each node has information only about its direct neighbors. We randomly pick up a node  $n$  and at each iteration of the algorithm, we measure the dissemination degree of its location information. The dissemination degree is estimated as the age of  $n$ 's location information averaged over all nodes in the network.

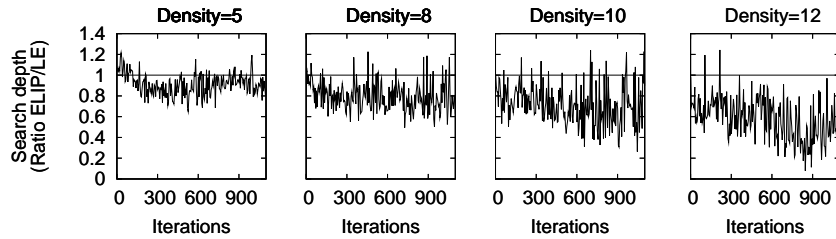
We can see in Fig. 4 that, whatever the distance, the average age given by ELIP is upper bounded (with a factor of two) by the result of LE. This figure also shows that, as expected, using ELIP instead of LE allows distant nodes to have fresher information about  $n$ 's position.

<sup>3</sup> The simulator is available at <http://www-spi.lip6.fr/~mandel/rml/simulator>.



**Fig. 4.** Average age of  $n$ 's coordinates depending on distance between  $n$  and the other nodes. The network density for this simulation is fixed to 8 nodes per coverage zone.

**4.2.2 Search depth** As described in Section 2.2, when anchor node  $a_i$  receives a message, it searches around its own position a node having fresher information about the destination than the one carried by the packet. The search depth is the distance, in number of hops, separating the anchor node and the node that responds to the local flood. In our simulations, we use EASE as the lookup algorithm for both LE and ELIP with different topology densities. Fig. 5 shows the ratio between current overheads generated by ELIP and LE. Observe that the ratio is almost all the time inferior to 1. This ratio varies between 0.5, when density is 12, and 0.9 for density 5.



**Fig. 5.** Average search depth, which represents the distance in number of hops between the anchor and the node which responds with a better estimation of destination's location.

**4.2.3 Cumulative overhead** For LE, traffic overhead is generated only when anchor nodes search for a better estimation of the destination's position. In ELIP, we also generate traffic overhead in data packets. It is then important to prove that, although ELIP incurs some overhead during the dissemination, the overall overhead, *i.e.* dissemination + lookup, is reduced.

Fig. 6 shows the cumulative overhead generated during the simulation. In the beginning, when the topology density is low, ELIP generates more overhead. The reason for this is that the position tables are about the same in both ELIP and LE cases. Thus, lookups lead to similar overheads. However, the overall overhead generated by ELIP is higher because of the extra overhead generated during the dissemination phase. But this overhead is rapidly compensated as data packets travel around.

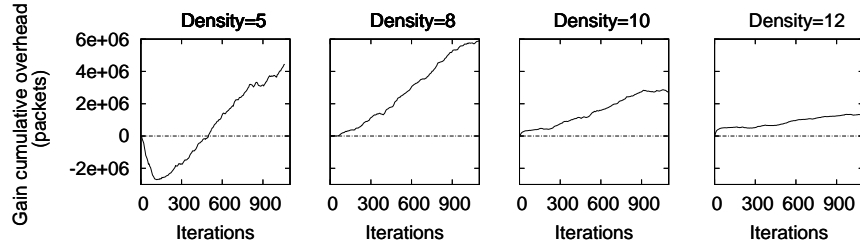


Fig. 6. Total overhead as a function of time.

**4.2.4 Path length** We also performed a set of simulations in order to evaluate the routes computed when using both LE and ELIP. We can see in Fig. 7 that routes computed when using ELIP are shorter. The difference between route lengths when using ELIP and LE grows with increasing densities. We see in the same figure that, when density is 5, ELIP is closer to LE. The routes obtained by ELIP are 50% shorter when density is 12 nodes per coverage region.

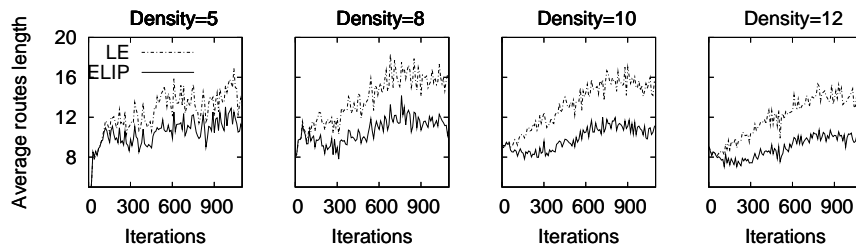


Fig. 7. Route length as a function of density.

## 5 Conclusion

This paper proposes *Embedded Location Information Protocol*, a dissemination method for age and position based routing algorithms. Existing approaches to locate nodes in ad hoc networks justify our choice for a database-free method. In this context, ELIP is an algorithm that uses existing data packets to perform dissemination in a very efficient fashion. Through a number of simulations, we could determine the effectiveness of our approach. The results show that the performance of ELIP is always better than the simple Last Encounter method. The reason for this is that ELIP, although increasing the overhead during the dissemination phase, drastically reduces the global overhead by allowing nodes to perform lookups in smaller scopes. Future improvements of ELIP include an overwriting model for the dissemination of a larger number of location information per  $T$  field and an insertion probability which depends on nodes speed and mobility model.

## References

- [1] Capkun, S., Hamdi, M., Hubaux, J.P.: GPS-free positioning in mobile ad hoc networks. In: Proceedings of the 34th HICSS. (2001)
- [2] Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications* **7** (2000) 28–34
- [3] Hightower, J., Borriella, G.: Location systems for ubiquitous computing. *IEEE Computer Magazine* **34** (2001) 57–66
- [4] Kranakis, E., Singh, H., J.: Compass routing on geometric networks. In: Proceedings of 11 th Canadian Conference on Computational Geometry. (1999) 51–54
- [5] Basagni, S., Chlamtac, I., Syrotiuk, V.R., Woodward, B.A.: A distance routing effect algorithm for mobility DREAM). In: Proceedings of ACM MOBICOM'98, Dallas, TX (1998) 76–84
- [6] Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks. In: Proceedings of ACM MOBICOM'00. (2000)
- [7] Mauve, M., Widmer, J., Hartenstein, H.: A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine* **15** (2001) 30–39
- [8] Grossglauser, M., Vetterli, M.: Locating nodes with EASE: Last encounter routing in ad hoc networks through mobility diffusion. In: Proceedings of IEEE Infocom. (2003)
- [9] Johnson, D.B., Maltz, D.A.: DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. In: Ad Hoc Networking. Addison-Wesley. (2001) 139–172
- [10] Perkins, C.E., Royer, E.M.: Ad hoc on demand distance vector (AODV) routing. In: Proceedings of IEEE Wrokshop Mobile Computing systems and Applications (WMCSA'99). (1999)
- [11] Viana, A.C., Amorim, M.D., Fdida, S., Rezende, J.F.: Indirect routing using distributed location information. to appear in *ACM Wireless Networks* (2004)
- [12] Eriksson, J., Faloutsos, M., Krishnamurthy, S.: Scalable ad hoc routing: The case for dynamic addressing. In: Proceedings of IEEE Infocom. (2004)
- [13] Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). Internet RFC 3626 (2003)
- [14] Mandel, L., Pouzet, M.: (ReactiveML) <http://www-spi.lip6.fr/~mandel/rml>.