Packets Disseminating Locations: Piggybacking the Inherent Mobility of Data

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 more than an inconvenient. 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, Packets Disseminating Locations (PDL), allows nodes to piggyback existing data packets to disseminate nodes' positions in the network. Contrary to traditional approaches that depend on encounters between nodes, PDL converges much faster and does not require permanent node mobility.**

***Index Terms*—System design.**

I. Introduction

In large scale mobile ad hoc networks, position-based routing has proven to be efficient because of its relatively 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:

- *Positioning*: each node determines its own position using an absolute positioning system like GPS (Global Positioning System) or a relative positioning algorithm like GPS-Free [1] or GPS-Less [2]. A survey of these methods can be found in [3].
- *Locating*: the network must provide a location service that allows a source to discover the destination's coordinates. Location services are composed of two phases: dissemination of location information and lookup operation.
- *Forwarding*: routing decision is based on the destination's position contained in the packet and on the position of the forwarding node's neighbors. Examples are the Compass routing [4] or Greedy Perimeter Stateless Routing (GPSR) [5]. Refer to [6] for further detail on such routing methods.

Both the positioning and the forwarding policies are crucial components of a routing architecture. Nevertheless, neither the positioning system nor the forwarding algorithm can provide by themselves an adequate solution if an efficient location service is not supplied. Some networks use dedicated and well known servers that centralize location information. In spontaneous and mobile ad hoc networks, such an approach is difficult to implement because there is no *a priori* knowledge of the server location itself. Furthermore, centralized solutions are not robust to network partitioning or to an eventual failure/departure of the location server. To overcome such limitations, some works have proposed the use of decentralized techniques. Most of them rely on flooding or distributed hash tables (DHT) [7]. Flooding based techniques consist of broadcasting the network until the destination responds with a message containing its current position.[1] This generates high traffic control overhead and is clearly not scalable. DHTs have been proposed as an alternative solution to appropriately distribute locations among nodes in the network [8], [9]. Although DHT-based approaches are fair and result in low lookup overhead, they are not robust to mobility [7].

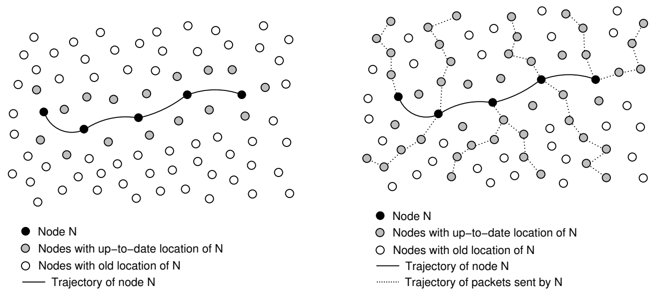
In this paper, we focus on the *dissemination* model associated to the location service. The dissemination model determines the degree of location information replication throughout the network. This replication may range from

[1] Of course some other node that receives the request may respond if it knows the destination's location. The problem in this case is to estimate how accurate this information is.

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 every other node in the topology. Each location is associated with an *age*, which gives the time elapsed since the last time it 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 [10]. 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 [10], 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 II and IV, 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 propose *Packets Disseminating Location* (PDL), an algorithm that uses *existing* data packets to disseminate location information. PDL relies on the basic assumption that packets are much more mobile than nodes. PDL piggybacks existing data packets and their inherent mobility, and incurs little traffic overhead to better disseminate topology knowledge than encounter methods. We will show that, although PDL incurs little overhead for disseminating location information, it largely reduces the global overhead (dissemination+lookup) when compared to the encounter approach. In Fig. 1, we show an example where a node N moves and disseminates its coordinates using both PDL and encounter-based method. Fig. 1(a) shows the scenario where nodes update N's location information using simple encounters. In Fig. 1(b), the same scenario is depicted when N uses packet mobility to disseminate its location information.



(a) Dissemination by encounters. (b) Dissemination in PDL.

Fig. 1. Dissemination models.

Our algorithm is efficient for these three reasons: (1) a node may communicate 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 PDL instead of encounters is much more efficient, mainly concerning the average age of location information in the network. Another important result is that the number of known destinations increases faster in PDL than in encounter-based methods. The consequence is that the lookup phase generates lower discovery traffic overhead and reduced delay. We also show through a number of simulations that the global overhead in PDL is upper bounded by encounter-based approaches and the gain increases as the network evolves.

The remainder of this paper is organized as follows. In Section II, we present the basic idea of the PDL algorithm. Section III details the main components of PDL and how nodes use existing data packets to disseminate location information. In Section IV, we evaluate the performance of our proposal with different topology sizes and mobility models, and compare it with the encounter-based approach. Finally, Section V concludes the paper and provides some ideas for future work.

II. PACKETS DISSEMINATING LOCATIONS: CONTEXT

Location is a crucial step in mobile ad hoc networks. It is even more important in position-based routing protocols because most of the overhead in such protocols comes from the location service, since both the positioning and forwarding steps are straightforward. The location service can be divided into two components: dissemination and lookup models. The dissemination model refers to the ability of the location service to distribute location information in the network. Our proposal, PDL, 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 and in Subsection II-B we present the lookup algorithm we will use to evaluate PDL.

A. 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. Excluding centralized approaches, we can classify location service strategies in four main groups:

- All-one: Every node knows only its own position. This method does not need any database maintenance because each node always knows its current position, but needs, when looking for a destination, to broadcast the network until the concerned node responds with its coordinates. This is clearly not scalable and generates high traffic control overhead. Successful protocols, not necessarily based on geographic coordinates, use such an approach, like DSR [11] and AODV [12].
- Some-some: The whole topology information is distributed among a subset of the nodes. When a node looks for a destination, it queries one of the servers, which responds with the destination's position. This approach, although simple, is considered unfair in the ad hoc concept because some nodes have more responsibilities than others [6].
- All-some: Every node in the topology plays the role of a rendezvous point. This category includes the DHT-based location service, where a node n stores its location information in a rendezvous node r depending on the n's identifier [9], [8]. When m wants to communicate with n, it sends a request to r which responds with the n's position. 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 changing nature of the topology, this may lead to high signaling overhead.
- All-all: Each node always knows the positions of all the nodes in the topology in a proactive fashion [13]. On the one hand, such an approach generates much traffic overhead in order to keep databases constantly updated. On the other hand, discovering a node's position is fast and does not generate any traffic overhead.

B. Age-and-position based routing (APB)

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. When n addresses a message to m, each node participating in the forwarding procedure may "deviate" the message if it has a fresher information about m.

The Last Encounter Routing (LER) has been proposed by Grossglauser and Vetterli as an implementation of the APB algorithm [10]. 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. We assume w.l.g. that two nodes have encountered each other if they have been directly connected in the past. In [10], the authors implement LER by using Last Encounters (LE) as the dissemination algorithm and Exponential Age SEarch (EASE) as the lookup method.

We briefly describe now EASE algorithm. For further details, please refer to [10]. Consider a node s that wants to communicate with d. Recall that, 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.

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.[2] The geographically closest node to $l_{s,d}$ is called an *anchor node*. Let anchor nodes be denoted by a_1, a_2, a_3, $\cdots$, and a_1 be the closest node to $l_{s,d}$. Node a_1 looks into its position table and if $\tau_{a_1,d} < \tau_{s,d}$ then a_1 sends the packet towards $l_{a_1,d}$. Otherwise, it locally floods a d's *position request* with some TTL (time-to-live) which determines the scope of the search. Node a_1 receives a response only if at least one of the nodes within this scope has encountered d more recently than s. If a_1 does not receive any

[2] 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.

response, it increases the TTL and floods a second request. This procedure is repeated until node n_1, the kth-degree neighbor of a_1, responds with $l_{n_1,d}$, *i.e.* the location of its last encounter with d, and $\tau_{n_1,d}$. Clearly, the inequality $\tau_{n_1,d} < \tau_{s,d}$ must hold. Node a_1 forwards then the message towards $l_{n_1,d}$, which will be received by the next anchor node a_2. The same search procedure is performed by a_2, and so on until the destination d is reached. It has been shown in [10] that the destination is reached after a number of steps of exponentially decreasing distances for a network with random node mobility.

III. PACKETS DISSEMINATING LOCATIONS: ALGORITHM DETAILS

Recall that the PDL algorithm is designed for ad hoc environments where nodes are supposed to know their geographic coordinates using GPS or any relative positioning system. The goal of PDL is to increase the dissemination degree of node location information, making it easier for a source to locate a destination. The originality of PDL is its ability to widely disseminate location information and reduce the global overhead by piggybacking existing data packets with position and age information of mobile nodes, and without creating any new signaling packets.

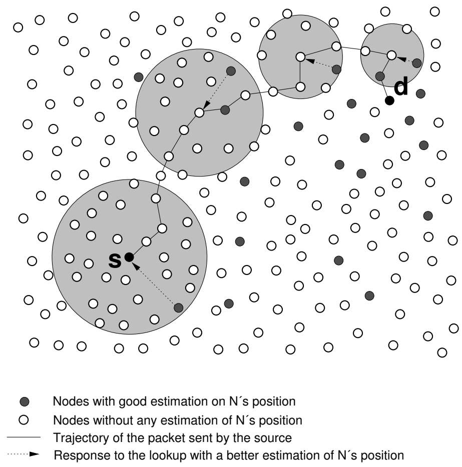
A. Node mobility $\times$ Packet mobility

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.

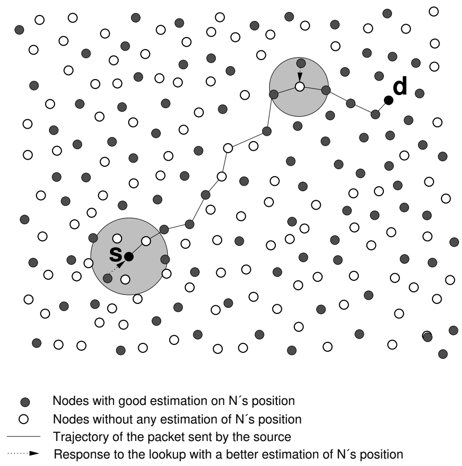
In PDL, location dissemination is implemented by encounters between nodes and by including location information in data packets. Because a node communicates with several other nodes, data packets cover a larger area through different routes. All nodes in these routes are then able to update their routing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

Fig. 2 shows an example of the functioning of EASE in two different scenarios. In Fig. 2(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. 2(b) can be achieved if nodes use PDL because updated location information is carried by packets, instead of



(a) Low dissemination.



(b) High dissemination.

Fig. 2. Impact of the dissemination model on the lookup overhead.

nodes. This is however difficult to obtain when using LE because nodes must permanently travel the entire topology in order to maintain a good distribution of location information throughout the network. The gains obtained by PDL include: smaller latency for location discovery, limited overhead, and shorter end-to-end paths.

B. Overhead

An important metric to evaluate the performance of a location service is the generated overhead. We define the *global overhead* of the location service as the sum of the dissemination overhead and the lookup overhead. We will see that PDL, when compared to LE, introduces little overhead for disseminating location information but drastically reduces the global overhead by limiting the lookup overhead (cf. Section IV).

C. Algorithm

In PDL, 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 PDL 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. Subsection III-D) 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 Subsection III-D how i obtains these positions). An example of $\mathbf{P}_i$ is shown in Fig. 3. In this position table, the information node i has about j is $A_{i,j} = [\mathtt{ID}_j, \hat{l}_{i,j}, \tau_{i,j}]$, 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 $\hat{l}_{i,j}$).

Node $\mathtt{ID}$: i

Dest. ID	$\hat{l}_{i,\cdot}$	$\tau_{i,\cdot}$
1	$(\hat{x}_{i,1}, \hat{y}_{i,1})$	$\tau_{i,1}$
2	$(?,?)$	∞
$\vdots$	$\vdots$	$\vdots$
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	$(?,?)$	∞

Fig. 3. Example of a position table. For the sake of clarity, "$(?,?)$" and "∞" indicate that node i has no estimation of the corresponding node.

Existing data packets are encapsulated in PDL packets with the structure shown in Fig. 4. In this packet, d is the destination node, l_d is an estimation of d's position, τ_d is the age of this estimation, and Γ is the current location of some node in the path traversed by the packet. The node that fills Γ depends on the insertion criteria presented in Subsection III-D. Fields l_d and τ_d are required by the lookup algorithm, which is common to both PDL and LE. The only overhead introduced by PDL is the Γ field, which contains the ID and the location of the node that has written in the packet. To prove that PDL is more efficient than LE, we have to show that the overhead introduced by Γ is compensated by a smaller overhead generated during the lookup phase (cf. Section IV).

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 τ_d with $\tau_{s,d}$. The Γ 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 [10].

l_d	τ_d	Γ	Original Data Packet

Fig. 4. Packet structure.

2) Writing coordinates in a packet: Let $P_{s,d} = \{s, p^1_{s,d}, p^2_{s,d}, \ldots, p^k_{s,d}, d\}$ be the path between s and d, where k is the number of nodes in the path (excepting s and d), $s = p^0_{s,d}$, and $d = p^{k+1}_{s,d}$. Node $p^i_{s,d}$ has the possibility to disseminate its location information, $\Gamma_{p^i_{s,d}}$, to every node $p^j_{s,d}$, $i < j \leq k+1$, by writing it in the Γ field of the data packets. In this paper, we assume that Γ 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. PDL is also designed such that only a small fraction of data packets are filled with Γ, in order to keep the dissemination overhead small. The probability for a node of writing in a packet will be described in Subsection III-D.

3) Updating position tables: In PDL, 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 every other node. Second, nodes update their position tables using information carried in the Γ field of transit packets. Destination d as well as all nodes $p^j_{s,d}$, $i < j \leq k$, update the $\{\mathtt{ID} = p^i_{s,d}\}$ entry in their position table by reading the information written by $p^i_{s,d}$ in Γ. 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.

D. Insertion probability

In this section, we explain how a node makes the decision to include or not its coordinates in a data packet. Recall that we assume in this paper that Γ can contain only one location vector and that it cannot be overwritten. A node decides to include its own location vector in the field of a packet depending on a probability π. We propose in the following two variants to compute this probability.

- *Fixed probability*: π is a fixed and predefined probability for all nodes. This approach is simple to implement but may lead to limited improvements because it does not take into account important characteristics of the network topology. Because every node has the same probability to write in Γ field, and because this field can no more be overwritten, the first nodes on the route have more chances to disseminate their location information than the last nodes.

- *Adaptive probability*: π is calculated depending on the route length R, estimated locally. The packet traveling toward the destination contains an estimated d's position and its age. Call them l_d and τ_d. In each forwarding node i, R is a function of both l_i and $best(l_d)$, where $best(l_d) = l_d$ if $\tau_d < \tau_{i,d}$ and $best(l_d) = l_{d_i}$ otherwise. The bigger the R, the smaller the probability P_i for node i to insert its location information.

IV. Analysis

A. Simulator model

We have conceived a network simulator to evaluate the efficiency of PDL and compare it with LE. The simulator implements EASE as the lookup algorithm. We describe in the following the parameters and assumptions used in our simulations.

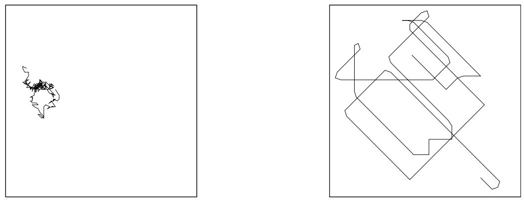
- *Topology.* The emulated network environment is a square universe of 1000-meter sides, partitioned into a grid with squares of one 1m^2. Vertices of the grid are the positions that nodes can occupy. The topologies are created with uniformly distributed nodes [14], where the number of nodes depends on the density we set for each experiment. This density, Δ, is the number of neighbors per coverage zone,

$$\Delta = \frac{N}{\sigma}\pi r^2, \quad (1)$$

  where N is the number of nodes in the topology, σ is the surface of the entire geographic region , and πr^2 is the coverage zone of the nodes.
- *Neighborhood.* Each node in the network has as immediate neighbors all nodes in a range of r meters. Two nodes i and j are neighbors if and only if $\delta(i, j)$, the Euclidean distance between them, respects

$$\delta(a, b) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r. \quad (2)$$

- *Mobility.* Node mobility is defined by two different random processes. Fig. 5(a) shows the path traversed by a node moving according to the pure random model. A similar picture is shown for the random waypoint model (Fig. 5(b)).



(a) Pure random model. (b) Random waypoint model.

Fig. 5. Examples of the two mobility models used in our analysis.

- *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 PDL for different values of traffic patterns, which is represented by the maximum number of simultaneous communications. 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 in our simulator that each node always knows its current coordinates and the ones of its direct neighbors. For forwarding, nodes 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 $N_i = \{n_1, n_2, ..., n_l\}$ be the set of i's neighbors. Node i forwards the packet to its neighbor n_x located in l_{n_x} such that $\delta(n_x, d) < \delta(i, d)$ and $\delta(n_x, d) \leq \delta(n_y, d), \forall n_y \in N_i$.

B. Measurements

We evaluate the performance of PDL in terms of the dissemination level. Because we compare PDL and LE methods, which are supposed to serve at the same layer, we do not take in account the interaction of the physical and MAC layers. We do not focus here on the performance of APB routing methods, but only on the advantages of using PDL instead of LE. In our simulator, each node has two position tables, one for LE and another for PDL. This guarantees that both algorithms are evaluated under identical conditions. For each route, we measure:

1) The number of known nodes: We consider in the beginning of our simulations that each node has information only about its direct neighbors. Thus, according to subsection IV-A, each node has in its position table, at this moment, only Δ entries. We assume that each position table can have up to N entries, *i.e.* the total number of nodes in the topology. Let k_i^t be the number of destinations known by node i at time t. The cumulative destination knowledge, K^t, is given then by

$$K^t = \sum_{i=1}^{N} k_i^t. \qquad (3)$$

Clearly, we have that $K^t \leq N^2$.

2) Average age of location information: As stated before, the lower the age, the better the estimation of a node's coordinates. The goal of PDL is to reduce as low as possible the age of the nodes' coordinates in the position tables. We have measured the average age of all nodes' position tables as a parameter to evaluate the diffusing degree obtained by LE and PDL. The global average age at instant t, G^t, is

$$G^t = \frac{\sum_{i=1}^{N} g_i^t}{N}, \qquad (4)$$

where g_i^t is the individual average information age at node i at instant t. The individual average age is given by

$$g_i^t = \frac{\sum_{i=1}^{k_i^t} \tau_{i,j}^t}{k_i^t}, \qquad (5)$$

where $\tau_{i,j}^t$ is the age of the information i has about j at time t.[3]

3) Local overhead: We use EASE as the lookup algorithm for both LE and PDL. Recall that in EASE, each anchor node searches around it fresher information about the location of destination. To perform this search, the anchor node locally floods a *destination position request* with increasing distances until a node responds with a better estimation of the destination's location. The current overhead is computed through the following measures:

- *Search degree:* As described in subsection II-B, when anchor node a_i receives a messages, it searches around its position a node that has in its position table a better information about the destination than the one carried by the packet. The search degree we measure is the distance, in number of hops, separating the anchor node and the node that responds the destination position request. Let ζ_{a_i} be the search degree at anchor node a_i. If packet p traverses l anchor nodes, then the search degree for routing this packet p from source to destination is

$$S_p = \sum_{i=1}^{l} \zeta_i. \qquad (6)$$

  It is also useful to define the average search degree for the whole topology at time t:

$$S^t = \frac{\sum_{i=1}^{k} S_i}{k}, \qquad (7)$$

  where k represents here the total number of packets sent in the topology at this time.

- *Total number of consulted neighbors:* It refers to the number of nodes participating in search. Let again l be the number of anchor nodes traversed by data packet p. Let also λ_{p,a_i} be the number of nodes reached by the destination position request flooded by the ith anchor node. The *total number of nodes participating in search* during the transmission of packet p at time t is

$$\lambda_p^t = \sum_{i=1}^{l} \lambda_{p,a_i}^t. \qquad (8)$$

  The *average number of consulted neighbors* at time t is given by

$$\Lambda^t = \frac{\sum_{i=1}^{k} \lambda_i^t}{k}, \qquad (9)$$

  where k is the total number of packets sent in the topology at this time.

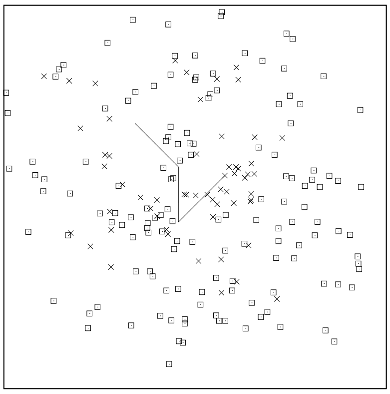
4) Cumulative overhead: For LE, traffic overhead is generated only when anchor nodes search for a better estimation of the destination's position. In PDL, we also generate traffic overhead in data packets. We show in the following section through a number of simulations that PDL significantly reduces the global overhead (diffusion+lookup) when compared with LE.

C. Simulation Results

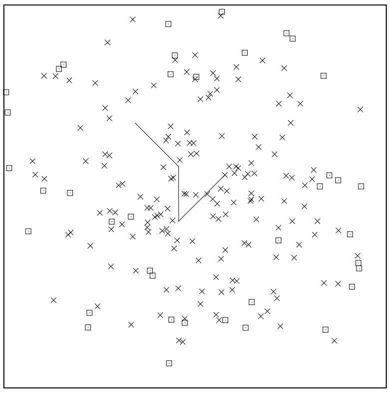
We have performed simulations to evaluate the efficiency of PDL in a square topology of 1000-meter sides, with a density of 10 nodes per coverage zone. The coverage range of the nodes is set to 90 meters and the mobility model used is the random waypoint model. Fig. 6 shows an example of the mobility models and the resulting diffusion degree for both LE (Fig. 6(a)) and PDL (Fig. 6(b)). In these pictures, the lines depict the path traveled by a node. Nodes represented by a box are nodes that do not have any information about the destination, while crosses represent nodes that have an estimation of the destination's coordinates. As we can observe from these pictures, PDL rapidly leads to a good dissemination level.

Fig. 7 shows the average age of location information of known destinations throughout the network. In the beginning, nodes know only their direct neighbors (age equal to zero), which explains the curves starting at the origin. We have evaluated PDL for two different values of traffic quantities, 20% and 30% (cf. Section IV-A). Note that even for a relatively small amount of data packets, PDL achieves better performance than LE, and that this difference increases as the network evolves.

[3]We take into account only the k_i^t known nodes to compute g_i^t.



(a) LE



(b) PDL

Fig. 6. The way followed by node n and the nodes aware of its coordinates.

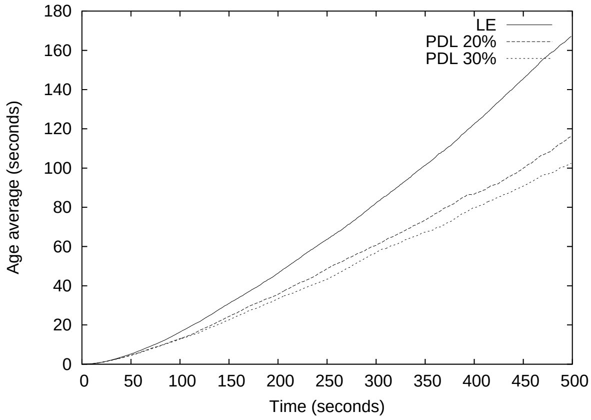


Fig. 7. Average age of known destination in nodes' position tables.

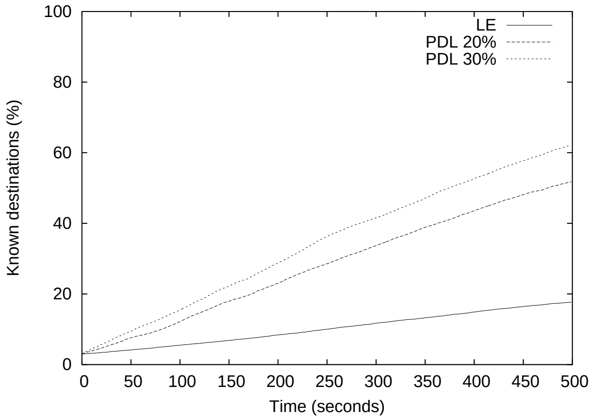


Fig. 8. Number of known nodes' coordinates as a function of time.

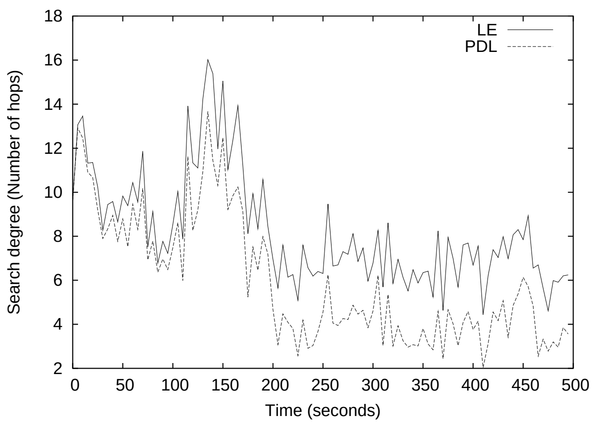
Another interesting parameter to observe is the absolute number of known destinations, over the entire topology. Using the same scenario of the previous experiment, Fig. 8 shows the results obtained for both LE and PDL. Once again, PDL shows better performance when compared with LE. These results confirm the expectation that packet mobility is much more efficient than node mobility for performing location dissemination.

We show in Fig. 9 the average search degree needed by anchor nodes to obtain fresher location information when the network uses either LE or PDL. Observe that, as shown in Fig. 9(a), the average search degree required by PDL is upper bounded by the one needed by LE, at any time. This means that the node which responds to the destination position request is closer to the anchor node when PDL is used. In the same experiment, Fig. 9(b) plots the average number of nodes participating in the lookup phase. Again, the PDL curve is always upper bounded by the LE curve. This result is due to the fact that when using PDL, anchor nodes send less position requests to get a best estimation about destination's coordinates than those included in the packet.

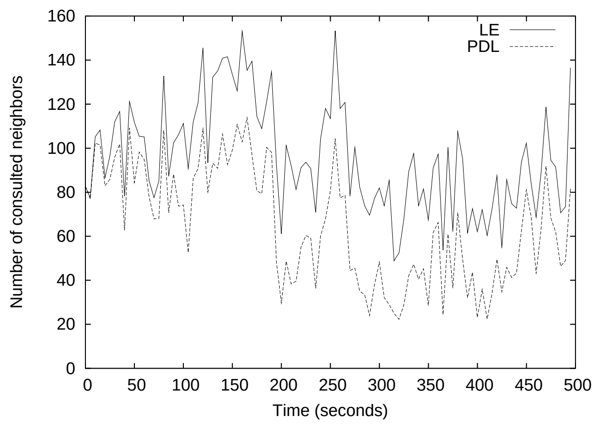
Finally, we present in Fig. 10 the efficiency of both PDL and LE in terms of global overhead. This confirms our expectation that, in spite of inserting coordinates in data packets, the global traffic overhead generated by PDL is lower than the one generated by LE, even when the probability of PDL insertion is relatively high. The reason for this is that the overhead generated by PDL during location dissemination is compensated by smaller search disks when anchor nodes look for a better destination's location.

V. Conclusion

This paper proposes *Packet Locating nodes*, 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, PDL 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 PDL is always better than the sim-



(a) Search disc degree.



(b) Number of consulted neighbors.

Fig. 9. Local traffic overhead at anchor nodes represented by the search degree and the number of consulted neighbors.

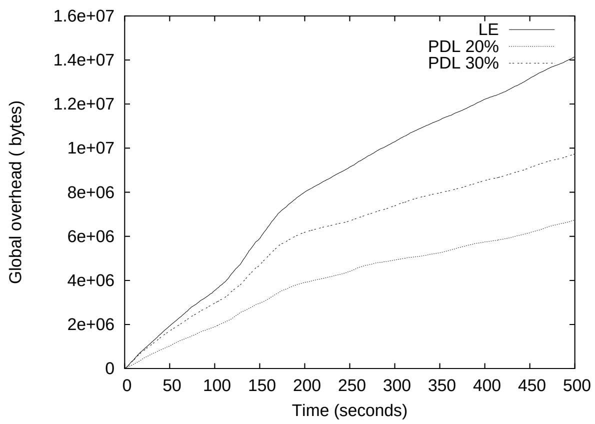


Fig. 10. Traffic overhead growth as a function of time.

ple Last Encounter method. The reason for this is that PDL, 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 PDL include an overwriting model and the dissemination of a larger number of location information per Γ field.

REFERENCES

[1] S. Capkun, M. Hamdi, and Jean-Pierre Hubaux, "Gps-free positioning in mobile ad hoc networks," in *Proceedings of the 34th HICSS*, Jan. 2001.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[3] Jeffrey Hightower and Gaetano Borriella, "Location systems for ubiquitous computing," *IEEE Computer Magazine*, vol. 34, no. 8, pp. 57–66, 2001.

[4] E. Kranakis, H. Singh, and J., "Compass routing on geometric networks," in *Proceedings of 11 th Canadian Conference on Computational Geometry*, Aug. 1999, pp. 51–54.

[5] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of ACM MOBICOM'00*, Aug. 2000.

[6] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, Nov. 2001.

[7] A. C. Viana, M. D. Amorim, S. Fdida, and J. F. Rezende, "Self-organization in spontaneous networks: the approach of dht-based routing protocols," *To appear in Ad Hoc Networks Journal*, Dec. 2004.

[8] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Scalable ad hoc routing: The case for dynamic addressing," in *Proceedings of IEEE Infocom*, Mar. 2004.

[9] A. C. Viana, M. D. Amorim, S. Fdida, and J. F. Rezende, "Indirect routing using distributed location information," *To appear in ACM Wireless Networks on december*, Dec. 2004.

[10] M. Grossglauser and M. Vetterli, "Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion," in *Proceedings of IEEE Infocom*, Mar. 2003.

[11] D. B. Johnson and D. A. Maltz, *Ad Hoc Networking*, chapter DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pp. 139–172, Addison-Wesley., 2001.

[12] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector (aodv) routing," in *Proceedings of IEEE Wrokshop Mobile Computing systems and Applications (WMCSA'99)*, Feb. 1999.

[13] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," *Internet RFC 3626*, Oct. 2003.

[14] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 1996.