Received 1 February 2011
Revised 18 May 2011
Accepted 14 June 2011

Proactive neighbor localization based on distributed geographic table

Marco Picone, Michele Amoretti and Francesco Zanichelli
Distributed Systems Group, Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Parma, Parma, Italy

Abstract

Purpose – A large set of valuable applications, ranging from social networking to ambient intelligence, may see their effectiveness and appeal improved when supported by the large-scale, real-time tracking of mobile devices, either carried by humans or embedded into vehicles. A centralized approach, where few servers would collect position data and provide them to interested consumers, would hardly cope with the resource demand of the foreseen huge increase of users interested in location-based services and with the flexibility needs of emerging user-generated services. The purpose of this paper is to propose a decentralized peer-to-peer approach to cope with these requirements, for which positioning information flows directly among mobile devices incurring in limited data exchange.

Design/methodology/approach – The authors propose a decentralized peer-to-peer approach for which positioning information flows directly among mobile devices incurring limited data exchange. A peer-to-peer overlay scheme is introduced called distributed geographic table (DGT), where each participant can effectively retrieve node or resource information (data or service) located near any chosen geographic position. Next, the authors describe a DGT-based localization protocol that allows each peer to proactively discover and track all peers that are geographically near to itself.

Findings – The authors provide a performance analysis of the protocol by simulating several 1,000 users that move across an urban area according to realistic mobility models. The results show that the solution is effective, robust, scalable and highly adaptable to different application scenarios.

Originality/value – The new contributions of this paper are a general framework called DGT, which defines a peer-to-peer strategy for mobile node localization, and a particular instance of the DGT that supports applications in which every node requires to be constantly updated about the location of its neighbors.

Keywords Peer-to-peer, Neighbor position discovery, Localization, Mobile computing, Location-based services, Information retrieval, Information services, Distributed systems

Paper type Research paper

1. Introduction

Real-time localization services are gaining more and more importance for a broad range of applications, such as road/highway monitoring, emergency management, social networking, and advertising. For example, Google Latitude (www.google.com/latitude) detects the location of the user (by means of WiFi, 2G/3G/4G mobile or global positioning system (GPS) satellite signals), and allows for sharing it with authorized contacts, thus anyone can be aware of the location of his/her friends. In such a system, however, two major drawbacks can be found. The first one is technical, i.e. all information is managed by centralized servers (Xiaolei *et al.*, 2009), that make the service not massively scalable and very expensive for a single service provider. The second issue is more "social": the service does not enable users to select a region and discover

International Journal of Pervasive Computing and Communications
Vol. 7 No. 3, 2011
pp. 240-263

1742-7371
DOI 10.1108/17427371111173022

all the people that are located there, thus preventing for example the establishment of new social relationships, or the dissemination of service advertisements.

Provided that users have to explicitly agree on allowing their localization and tracking, we foresee location-based services (LBS) supported by an infrastructure implemented as a network of decentralized software entities, either with flat or hierarchical organization. An example of such LBSs may be a traffic monitoring application spread over a set of nodes placed along highways, each one collecting information about local traffic, and being able to provide aggregated information (e.g. statistics) to any remote user that requests it. Any centralized approach would hardly scale for such highly variable contexts, with mobile nodes changing their location very quickly, in a wide and highly populated area. Indeed, search results would be incomplete or outdated with high probability. Like other researchers (see Section 2) we believe that a partially or fully decentralized approach can increase the accuracy of information and the rate at which they are retrieved by users. Moreover, it may allow to update and publish information directly, with low-cost, and high-scalability. Last but not least, it may simplify the process of joining the virtual community and the implementation of collaborative LBS.

In this paper, we describe a general framework called distributed geographic table (DGT), that defines a peer-to-peer strategy for mobile node localization, and a particular instance that supports applications in which every node requires to be constantly updated about its neighbors. Compared to centralized approaches, the DGT is less expensive (because resources are provided by end-users, according to a participatory model) and more scalable (because its performance in terms of responsiveness, completeness, and robustness remains valuable also for a large number of nodes and when dynamics are very high).

The rest of the paper is organized as follows. Section 2 presents a survey and discussion of state-of-art research on location based search in peer-to-peer networks, outlining space representation techniques. Section 3 introduces the main concepts of DGT, including a formal definition of neighborhood, as well as the functional specifications of routing and maintenance strategies. Section 4 describes the peer neighborhood construction protocol we have implemented, based on DGT principles. Section 5 illustrates five different simulated scenarios we have implemented in order to evaluate the performance of the DGT-based localization protocol, for different configurations of parameters. Finally, we conclude the paper with a summary of the contributions of this paper and with some ideas for future work.

2. Related work

In recent times, the need for collaborative work supported by geospatial data has escalated, also due to events such as terrorist activities and natural disasters. An example of geocollaboration service is distributed localization, that is usually implemented by recursively dividing the 2D space into smaller areas, and assigning the responsibility of each one to a peer. Instead of employing a number of centralized servers (either dedicated or selected among participating nodes) to carry the information load for the entire network, this approach is much scalable and robust. New peers may join and support other peers, triggering a redistribution of the spatial responsibilities. Peers are networked, forming a mesh and contributing to message propagation over there (e.g. queries). The idea of hierarchical partitioning comes from the indexing techniques of data structures for multidimensional data-sets used in centralized database, such as R-tree (Guttman, 1984).

Examples of general-purpose hierarchical peer-to-peer schemes supporting geocollaboration services are HZSearch (Tran and Nguyen, 2008), DPTree (Li *et al.*, 2006), DiST (Namand and Sussman, 2006). These and other works (Tanin *et al.*, 2007; Liu *et al.*, 2005; Harwood and Tanin, 2003) propose strategies for supporting complex queries over multidimensional data, such as "select five available buildings closest to the airport". The specific problem of geographic localization is addressed for example by Geographical LOcation BAsed SEarch (Globase.KOM) (Kovacevic *et al.*, 2007), which adopts a tree-based peer-to-peer overlay enhanced with interconnections. The Globase.KOM scheme is based on supernodes, i.e. powerful nodes, with best network connectivity, that tends to stay online for a long time. Supernodes are responsible for indexing all nodes/services in one clearly defined geographical area. Other nodes in the network simply offer and consume services without having additional responsibilities. The idea is that the world projection is divided into disjoint, non-overlapping zones. Each zone is assigned to a supernode (located inside the zone itself) that has to collect, store, and maintain the overlay/underlay contact addresses of all nodes in that zone. Supernodes form a tree, where node A is called the parent of node B when B's zone is inside A's zone. Another architecture, called GeoP2P (Asaduzzaman and Bochmann, 2009), still performs a hierarchical partitioning of the 2D geographic space, but adopts a fully decentralized peer-to-peer overlay scheme, with overlay maintenance and query routing performed without super or special peers.

The main drawback of the hierarchical approach is that peers representing higher level regions may become bottlenecks for query routing, and possible points of failure for the whole system. Moreover, none of the state-of-art solutions has been demonstrated to work in presence of mobile peers.

Recently, an egalitarian approach has been proposed by Ekambaram and Ramchandran (2010), where all nodes collaborate and help each other to refine their position estimates, running distributed algorithms in the interest of scalability and reduced computational complexity. The proposed approach assumes the inclusion of static "anchor" nodes with known locations, that improve the system performance but introduce a cost for the municipality.

On the contrary, our DGT framework that adopts an egalitarian peer-to-peer model without infrastructural elements takes into account mobile nodes, thus enabling disruption-tolerant networks (Balasubramanian *et al.*, 2007). DGT is based on the principle that different geospatial applications have different information needs, for which it is not fair to constrain the framework with a unique data management strategy. For example, the DGT-based protocol for peer neighborhood construction, described in Section 4, adopts a proactive data collection strategy that makes it highly suitable to address situations where each user must be always aware of which nodes are in his/her surroundings.

3. Distributed geographic table

A decentralized peer-to-peer overlay is structured (i.e. based on the decentralized structured model (DSM)) if its topology is controlled and shaped in a way that resources (or resource advertisements) are placed at appropriate locations (Amoretti, 2009).

DGT is a structured overlay scheme where each participant can efficiently retrieve node or resource information (data or services) located near any chosen geographic position. In such a system, the responsibility for maintaining information about the position of active peers is distributed among nodes, for which a change in the set of participants causes a minimal amount of disruption.

In the following of this section, we present the main DGT concepts, using the peer-to-peer system notation introduced by Aberer *et al.* (2005).

3.1 Conceptual model and neighborhood

In a generic DGT overlay, the set of peers is called $\wp$, each peer having a unique $id \in \mathcal{I}$ (where $\mathcal{I}$ is the space of identifiers), and a pair $\langle latitude, langitude \rangle$. The space of world's coordinates is called $\mathcal{W}$ and $w \in \mathcal{W}, w = \langle latitude, langitude \rangle$ is the generic location.

Thus, a peer $p \in \wp$ may be identified by the pair $\langle id_p, w_p \rangle$, where $id_p \in \mathcal{I}$ and $w_p \in \mathcal{W}$. The association between a peer and an identifier is established by function $F_{\mathrm{p}} : \wp \to \mathcal{I}$.

In a DGT, the distance between two nodes is defined as the actual geographic distance between their locations in the world (also known as great-circle distance or orthodromic distance):

$$d : \mathcal{W} \times \mathcal{W} \to \mathbb{R} \tag{1}$$

The neighborhood of a geographic location is the group of nodes located inside a given region surrounding that location. More precisely, given the set of all geographic regions delimited by a closed curve $\mathcal{A}$, neighborhood is defined as:

$$N : \mathcal{W} \times \mathcal{A} \to 2^{\wp} \tag{2}$$

where $2^{\wp}$ is the set of all possible connections between peers. In order to evaluate the neighborhood of a target geographic point, $t \in \mathcal{W}$ using a region $a_{\mathrm{t}} \in \mathcal{A}$ centered in t, let us define:

$$\mathcal{N} = \{p \in \wp | w_p \subseteq a_{\mathrm{t}}\} t \in \mathcal{W}, a \in \mathcal{A} \tag{3}$$

where, as earlier, w_p is the geographic position of peer $p \in \wp$. By selecting for example, a circular region $C \in \mathcal{A}$, with a radius $c_r \in \mathbb{R}$, it is quite simple to evaluate the node's neighborhood:

$$c_t = \{w \in \mathcal{W} | d(w, t) \le c_r\} t \in \mathcal{W} \tag{4}$$

$$\mathcal{N} = \{p \in \wp | w_p \subseteq c_t\} t \in \mathcal{W}, a \in \mathcal{A} \tag{5}$$

If p is moving, then $\mathcal{N}_p$ dynamically changes accordingly.

3.2 Routing strategy

The main service provided by the DGT overlay is to route requests to find available peers in a specific area, i.e. to determine the neighborhood $\mathcal{N}$ of a generic global position (GP) $w \in \mathcal{W}$.

Routing is a distributed process implemented as asynchronous message passing. By executing the *route(p,w,a)* operation, a peer forwards to another peer $p \in \wp$, a request for the list of nodes that peer p knows to be located in region $a \in \mathcal{A}$, whose center is $w \in \mathcal{W}$.

Thus, a routing strategy can be described by a possibly non deterministic function:

$$\mathcal{R} : \wp \times \mathcal{W} \times \mathcal{A} \rightarrow 2^{\wp} \tag{6}$$

that returns the neighborhood $\mathcal{N}(w,a)$, around the geographic position w and within region a, known by peer p.

The routing process is based on the evaluation of the region of interest centered in the target position. The idea is that each peer involved in the routing process selects, among its known neighbors, those that presumably know a large number of peers located inside or close to the chosen area centered in the target point. If a contacted node cannot find a match for the request, it does return a list of closest nodes, taken from its routing table. This procedure can be used both to maintain the peer's local neighborhood $\mathcal{N}$ and to find available nodes close to a generic target.

Regarding the local neighborhood, the general aim of the approach is to have high knowledge of nodes that are close to the peer and a gradually reduced number of known nodes that will be used to forward long range geographic queries. This idea recalls Granovetter's (1973) theory of weak ties, stating that our society is formed by small complete graphs whose nodes are strongly connected (friends, colleagues, etc.). These clusters are weakly connected between each others, e.g. a member of a group superficially knows a member of another group. The most important fact is that weak ties are those which make our society an egalitarian small world network, i.e. a giant cluster with small separation degree and no hubs. Our society is not the unique network whose topology can be described by the weak ties model. Other examples are: the electrical grid, the neural network, several transportation networks (roads, railroads, airports).

In a DGT-based overlay, each peer owns, and maintains a structured routing table whose organization is based on geographic distance. Such a table can be organized in different ways to support the selected routing algorithm and to reduce the number of message propagations necessary to obtain the requested results. A routing algorithm for Distributed Systems Group can be characterized by means of performance measures, such as the average number of propagations per message, the probability of successful routing and the percentage of discovered peers compared to the actual number of nodes that are present in a specific area.

3.3 Maintenance strategy

The participation of peers in an overlay network dynamically changes over time. DGT addresses mobile peers that frequently/continuously change their geographic position, and may also join and leave the network at any time. As a consequence, the neighborhood N of a peer p $\in$ P is usually characterized by high dynamism. To preserve the consistency of the DGT, each peer needs to periodically schedule, a maintenance procedure that compensates the topological changes of the network. The practical usability of a DGT critically depends on the messaging and computational overhead introduced by such a maintenance procedure, whose features and frequency of execution are application-dependent. A general approach to DGT maintenance is illustrated in Section 4.

3.4 Security

A very important issue in real-time localization concerns security and privacy. In DGT, the only data that are shared among peers are their unique identifiers, their internet protocol (IP) addresses and ports, as well as their GPS coordinates. At the DGT level, no reference exists to sensitive data (e.g. media access control address) that may allow to identify a node's owner. The capability of finding peers that are active and close to a specific geographic position is obtained without sharing any personal data. Sensitive or potentially dangerous information may be added by applications built over the DGT, that may store such data in their data structures, but this problem potentially affects any peer-to-peer scheme and has been addressed by several research works (Balfe *et al.*, 2005; Aringhieri *et al.*, 2006; Kesar, 2003; Kher *et al.*, 2005).

4. Peer neighborhood construction

Within the DGT framework presented in Section 3, we have designed a proactive peer-to-peer protocol for the localization of all nodes that are "geographically" (rather than "virtually") close. Indeed, whenever a single active node in the system wants to contact other peers in its area (e.g. to provide or search for a service), it does not need to route additional and specific discovery messages to its neighbors (or to a supernode responsible for a specific zone) in order to find peers that are geographically close. Instead, it simply reads its neighbor list, that is proactively filled with "geographic neighbors".

Our peer neighborhood construction protocol has been inspired by Kademlia (Maymounkov and Mazieres, 2002), a famous DSM-based protocol, used for example in recent versions of the eMule client (as an alternative to the traditional eDonkey protocol). Many of Kademlia's benefits result from its use of the XOR metric for distance between points in the key space. XOR is symmetric, allowing Kademlia participants to receive lookup queries from precisely the same distribution of nodes contained in their routing tables, that are organized as sets of "k-buckets". Every k-bucket is a list having up to k entries;, i.e. all nodes on the network have lists containing up to k nodes for a particular bit (a particular distance from itself). To locate nodes near a particular ID, Kademlia uses a single routing algorithm from start to finish. In contrast, other systems use one algorithm to get near the target ID and another for the final hops.

Being based on the DGT, our localization protocol uses the geographic metric, instead of Kademlia's XOR metric. Each node knows his GP retrieved with GPS system or with other localization technologies, and knows a set of real neighbors organized in a specific structure based on the distance that these nodes have in relation to the node's position.

The main goal of the protocol is to build and maintain an overlay where each node knows all the active nodes that are available in a geographic region, in order to provide and realize specific applications and services. An example of application based on such a protocol may be a city monitoring system that uses decentralized nodes to monitor the traffic status of the city. By using this system, there is no need to deploy powerful servers: light peers can be activated in strategic locations, in order to cover the whole city area. Each of them can analyze its region of interest, monitor traffic conditions in real-time, and evaluate the position of peers in order to inform them about accidents and traffic jams, suggesting alternative paths.

4.1 Data structures

Every peer maintains a set of geo-buckets (GB), each one being a (regularly updated) list of known peers sorted by their distance from the GP of the peer itself. GB can be represented as K concentric circles, each one having a different (application-specific) radius R_i and thickness r_i, being i an integer $\in [1,K]$, for which:

$$R_i = \sum_{j=1}^{i} r_j \tag{7}$$

If there is a known node whose distance from the peer is larger than the radius of the last circle , it is inserted in another list that contains the nodes outside the circle model.

Each peer in the GB set is characterized by:

- Unique ID – univocally identifies the peer within the DGT.
- GPS position – latitude and longitude retrieved with a GPS system or with other solutions (e.g. GSM cell-based localization).
- IP address – allowing to identify the node in internet; if the peer is behind network address translation, the IP address may be that of a relay.
- UDP port – on which the peer listens, waiting for connection attempts.
- Number of known nodes – used to compare two nodes that have the same distance.

4.2 Network join

When a new peer wants to enter the network, it sends a join request, together with its GP, to a bootstrap node, that returns a list of (up to L) references to peers that are geographically close to the joining one. It is important to emphasize that this information is not updated: referenced peers may have moved away from their initial location. It is up to the joining peer to check for the availability of listed peers. This operation is performed not only during the first join of the peer, but also when the peer finds itself to be almost or completely isolated. In these situations (that typically arise when peers enter low density areas), the node may send a new join request to the bootstrap node, in order to obtain a list of recently connected peers that may become neighbors.

4.3 Peer lookup

The main procedure used during peer discovery is *findNodes(GP)*, that returns the β peers that are nearest to the specified GP. By periodically applying *findNodes()* to GP_n, peer n keeps up to date its neighborhood awareness. Such a procedure (with any target GP) may also be executed upon request from another peer.

Node n searches in the GB associated to the requested GP. The final objective of the lookup (summarized in algorithm 1) is to find the $\alpha < K$ peers that are nearest to the selected GP, including newly connected nodes, as well as mobile peers that have entered the visibility zone. The lookup initiator starts by picking α nodes from its closest non-empty GB or, if that bucket has less than α entries, it just takes the α closest nodes, by extending the lookup to all its GB). Such a peer set is denoted as $C_i = \{n1i, \ldots, n\alpha i\}$, where i is an integer index. The initiator sends parallel findNodes requests, using its GP as target, to the α peers in C_i. Each questioned peer responds with β references. The initiator sorts the result list according to the distance from the target position, then picks up α peers that it has not yet queried and re-sends the *findNodes* request (with the same target) to them. If a round of *findNodes* fails to return a peer closer than the closest already known, the initiator re-sends the *findNodes* to K closest nodes it has not already queried. The lookup terminates when the initiator has obtained responses from the K closest nodes, or after f cycles, each cycle resulting with an updated set of nearest neighbors C_i. Thus, the number of sent (*findNodes(GP)*) messages is $f \cdot \alpha + K$, that depends on the density of peers in the area of interest. A peer is allowed to run a new lookup procedure only if the previous one is completed, in order to reduce the number of exchanged messages and avoid the overlapping of same type of operation.

4.4 Position update

Any peer (let us call it A) active in the network can change its geographic position, for many reasons (the user may be walking, driving, etc.). In order to maintain a good level of knowledge accuracy within the network, each node notifies its GP updates to neighbors. In order to reduce the computational and bandwidth impact of this operation, before sending a position update, peer A performs the following two operations, involving every node (called B, for simplicity) in its GB.

Peer A checks for the distance between itself and peer B: $d_{AB} = dist(w_A, w_B)$. If such distance is larger than R_K, it means that peer B is out of the visibility area of A, and for this reason, it removes B from its GB and sends to it a *RemoveMessage*, in order to notify the removing operation. This action is very important because, if peer B does not receive this specific message, it keeps the reference of A in its GB, but it does not receive new updates because A removed B from its GB (Figure 1):

Algorithm 1. periodicLookup(GP)

$i \leftarrow \mathbf{0}$
get nodes from geo-buckets (nearest to GP): $C_i = \{n_{1i}, \ldots, n_{\alpha i}\}$
repeat
  $j \leftarrow \mathbf{1}$
  while j ≤ α **do**
    if n_{ji} not yet queried **then**
      $n_{ji} \cdot findNodes(GP)$
    end if
    $j \leftarrow j + \mathbf{1}$
  end while
  get α nodes (nearest to GP) from the $\alpha\beta$ results: $C_{i+\mathbf{1}}$
  $i \leftarrow i + \mathbf{1}$
until $C_{i+1} = = C_i$

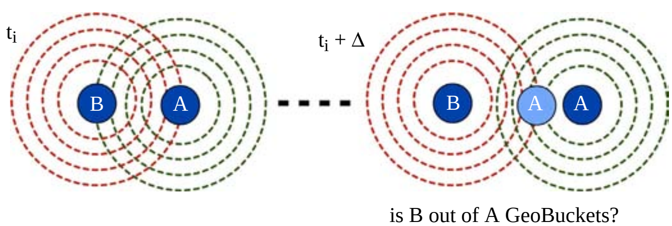

Figure 1. While moving, peer A finds peer B to be too much far, thus removes it from its GB, and notifies the operation B

```
f ← i
get K nodes (nearest to GP) from geo-buckets, not already in C_j
j ← i
while j ≤ K do
  if n_ji not yet queried then
    n_ji · findNodes(GP)
  end if
  j ← i + 1
end while
```

Peer A checks for $d(A) = d(W_{Anew}, W_{Aold}) > \varepsilon$. If the condition is true, A sends its position update to its neighbors. Parameter $d(A)$ allows to define the accuracy of update messages and can be configured according to application requirements. A small value of ε causes a high rate of exchanged messages, but a large value reduces the accuracy of the peer's knowledge and damages the global performance of the protocol.

There is another important aspect related to position updating. In order to improve the performance during the join procedure, each peer sends update message to the bootstrap node if the distance between the peer's actual position and the one that it had when it entered the network has becomes larger then λ, i.e. $d_b(n) = dist(GP_{boot}(n), GP_{new}(n)) > \lambda$. Such an update is performed only if the peer is moving far from its original area, and helps the bootstrap node to provide more precise information to newcomers (Figure 2).

4.5 Gossiping

To improve the results and the performance of our protocol, we added gossip information inside exchanged messages. This approach on one side increases the size of sent packets, but at the same time, adds significant knowledge that helps peers to be aware of available nodes in their area. Each peer maintains the references to nodes that have been discovered during the time between two different lookup procedures.

5. Performance evaluation

Evaluating the performance of our DGT-based protocol in significant, dynamic scenarios cannot be done analytically, because of the high complexity (non-linearity) of the problem. For this reason, we used the discrete event simulation tool called discrete event universal simulator (DEUS) (Amoretti *et al.*, 2009), that provides a simple Java application programming interfaces (API) for the implementation of nodes,

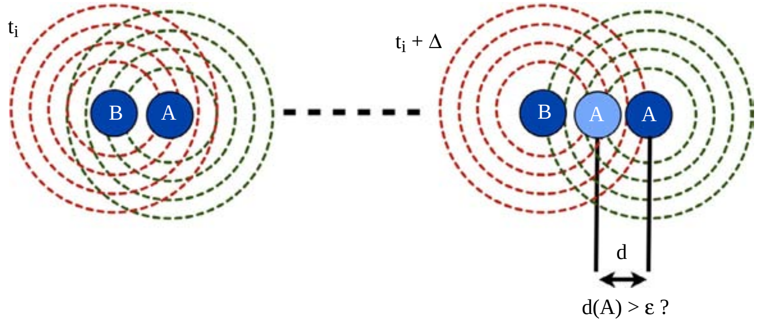

Figure 2.
After a movement peer A checks if it is necessary to notify the new position to its neighbors

events and scheduling processes, and a straightforward but powerful visual tool for configuring simulations of complex systems.

In order to perform realistic tests, we set up an integration between DEUS and Google Maps API. With the features provided by Google Maps API, we have created a simple HTML/Javascript control page that allows to monitor any simulated node, following it from starting to final position, and all the neighbors in its GB. This solution allows to study the protocol not only with specific peer-to-peer metrics like message rate (MR), miss ratio, number of peers, etc. but also with a direct monitoring of peer behaviors during the simulation.

5.1 Performance metrics

We took into account the following metrics, to evaluate the performance of our DGT-based peer neighborhood construction protocol:

- *PMN*. Percentage of missing nodes in the GB of a peer, with respect to those really present in the area.
- *MR* (*msg/sec*). Message rate, i.e. the average number of messages received per second by each node.
- *NPE* (*km*). Node position error, i.e. the average distance between a peer's position reference in a GB and its actual position.

5.2 Peer mobility model

In order to evaluate the performance and behavior of our DGT-based protocol, we considered two different mobility models, namely a very generic one and another specific to street vehicles. Two different metropolitan areas have been chosen for the simulation, the first one around Frankfurt (square area with side of 20 km) and the other surrounding Parma (square area with side of 7 km). In both cases, a list of real road paths have been generated offline (using the GoogleMaps API and a refining algorithm) from an initial set of potential points of interest.

The first mobility model is a random model where an active peer in the system selects one of the available paths and starts moving over it segment after segment. Each peer is a mobile node with a random base speed (V_b) between 5 and 100 km/h that can be associated to pedestrians, bikers, and vehicles. For each segment, peer speed is randomly selected according to an exponentially distributed random variable with mean value V_b.

The second mobility model, we considered is more complex also from a computational point of view, since it takes into account characteristics and parameters of inter-vehicular networks. Among available solutions and considering the chosen evaluation process the Fluid Traffic Model (Seskar *et al.*, 1992) has been adopted. It describes speed as a monotonically decreasing function of vehicular density, forcing lower speed when traffic congestion reaches a critical point taking into account car flow characteristics, queues, and traffic jams of an urban environment.

5.3 Evaluation of geo-bucket configuration

The first part of our simulation analysis aims at outlining how the choice of the GB configuration, in terms of number of GB and their thickness, influences DGT performance as expressed by the PMN and the MR. The following results refer to the generic mobility model, unless otherwise specified.

Two different peer systems are simulated over a significant time span. The first one includes 1,000 peers for a virtual time of ten hours (corresponding to 10,000 virtual time units) while the second one has doubled size (2,000 peers) and time span (20 hours, that is 20,000 virtual time units). In both cases, the node set grows to full size during the first half of the simulation, after which only an insignificant number of peers enters and leaves the network. Table I presents all the considered cases.

Figure 3 shows the PMN for cases 1, 2, and 3, which refer to the first peer system (1,000 nodes over ten hours). We remark that the PMN remains moderate over all the simulated period, for all GB configurations. In particular, for the first half of simulation, where many new nodes enter the system, the PMN value only grows up to around 5 percent, while it decreases significantly when the peer network reaches a more stable state.

Another important metric that we must take in account to evaluate these different configurations is the MR. Figure 4 shows results for cases 1, 2, and 3. Given that a larger covered area ($\pi \cdot r^2$) is potentially associated to a higher number of GB active peers, an increased number of known nodes must be contacted to obtain GP updates.

For this reason, simulation results show that cases 1 and 2 have an increased MR value compared with case 3, where the covered area is smaller. In any case, the number

Table I.
Configuration of simulated scenarios

Case	No. of geo-buckets	GB thickness (km)	No. of peers	Final virtual time
1	10	1.5	1,000	10,000
2	5	3	1,000	10,000
3	10	0.5	1,000	10,000
4	10	1.5	2,000	20,000
5	5	1.5	2,000	20,000

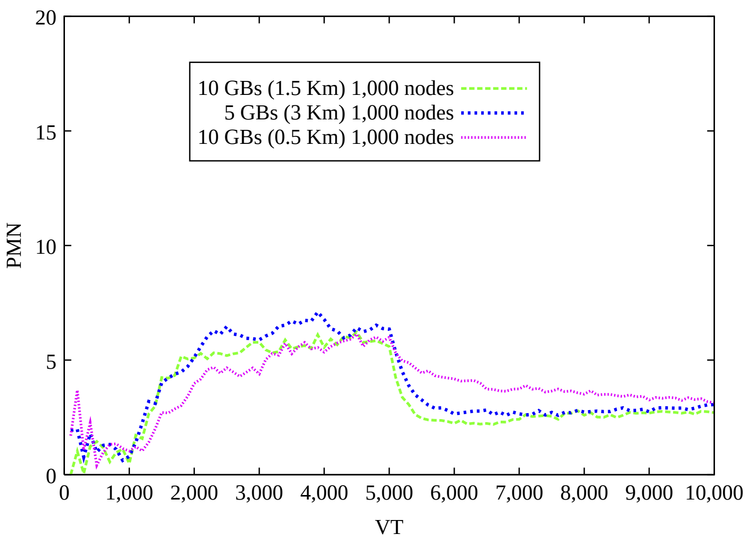

Figure 3.
PMN GB configuration evaluation (cases 1, 2 and 3) with 1,000 nodes

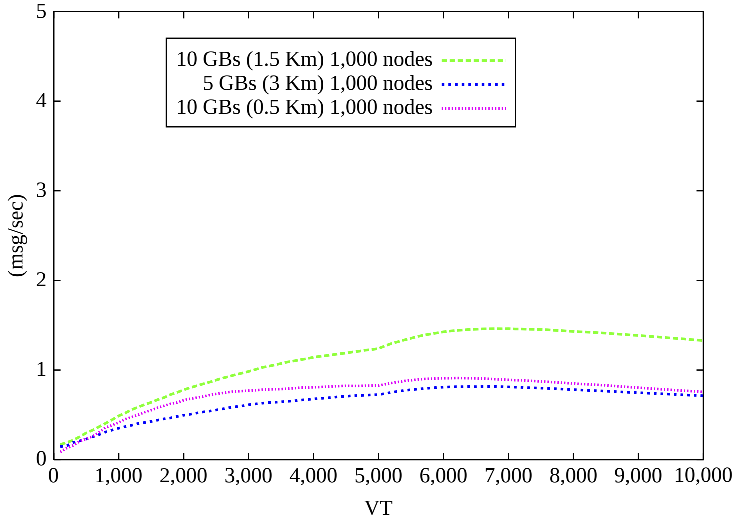

Figure 4.
Global MR GB configuration evaluation

of exchanged messages is very low, notwithstanding the fact this is a fully decentralized system where knowledge is maintained cooperatively by all available peers.

The same analysis was carried out on the larger (2,000 active peers) network using two configuration of GB (cases 4 and 5) in order to assess the protocol's behavior with a different distribution of nodes.

Results in Figure 5 show that also with a higher number of available peers and using two GB configurations the PMN is very small (under 10 percent and around 5 percent). In the fourth case, ten GB with 1.5 km thickness are used, which means a covered area of 706 km^2, whereas in case 5, we have only five GB with the same

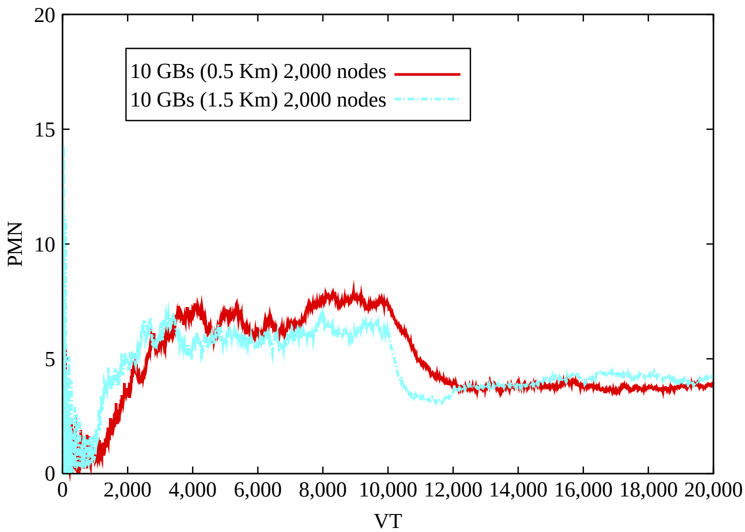

Figure 5.
PMN GB configuration evaluation (cases 4 and 5) with 2,000 nodes

thickness for a covered area of 176 km^2. There an evident difference in the covered area but the performance is very good in both cases. The little amount of missing nodes depends on the dynamics created by new incoming peers and by the high-rate of movements generated by nodes traveling on their paths.

In order to provide this level of performance with different configurations and covered areas, the protocol needs to route messages to users in the target zone. The denser scenario, as described for the smaller network, implies a different amount of exchanged messages (Figure 4) that depends on peer density in the analyzed area.

We observe that the accuracy of the protocol shows little dependence on the configuration of GB (number and thickness). Results show that different parameter setups still obtain very low PMN, given the highly dynamic context where all peers are mobile users that change their position very often. The other important aspect that comes from this analysis is the relationship between the covered area and the MR value, that we must take into account when designing an application based on this protocol, in order to find the right compromise between the size of analyzed zone and the number of exchanged messages.

Another important issue related to the PMN is to understand the distribution of missing nodes across available GB in order to verify the knowledge evolution of active peers.

Figure 6 is related to case 1, with ten GB having a 1.5 km thickness. We already showed the associated PMN in Figure 3 which stays very low for all the simulation at about 5 percent or less. We analyze now how this value is split across different GB. For the inner GB, the percentage of missing node is around 0 percent for the whole simulation's time.

Predictably, the largest amount of missing peers is located in the external GB that cover areas even very far from the peer. The outmost GB, i.e. GB9 has the highest PMN and other GB limit the PMN under the 20 percent. This is a very important result that shows how the protocol is very accurate and reliable and how it fulfills the DGT goal

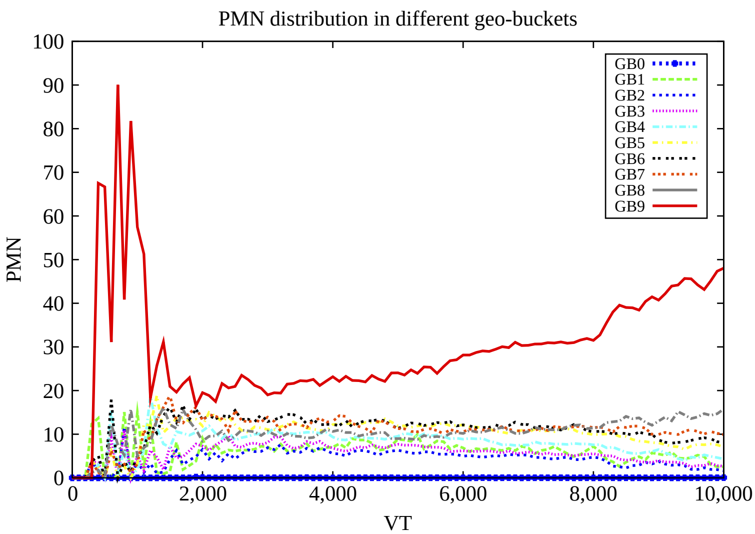

Figure 6.
PMN distribution in each geo-bucket GB configuration evaluation (case 1)

of having a high percentage of known peers that are very close to a node's position. This result was obtained with a 1.5 km GB thickness and may be very useful for applications like vehicular networks, where it is very important to have the best knowledge of active users in a specific area of interest around the car.

5.4 Larger network

The performance of our protocol was also evaluated in the context of an increased number of peers and with high dynamics due to the numerous joins. The simulation considered ≈ 5,400 peers over a virtual time of 50 hours using ten GB with 1.5 km thickness.

Figure 7 shows the achieved percentage of missing peers that appears slightly increased if compared with the results of the first scenario, although in any case it is reasonably under the 10 percent.

The cost in terms of exchanged messages (Figure 8) is still very low, if we consider that the GB covered area is large and the high density of active peers. We can see that in the first half of simulation, there is an increase of the analyzed parameter because there are a lot of new joins over a short time and in the same area. This behavior causes new activities related to joins and position updates that require additional message exchange among peers. In the second half, when the number of new incoming users is decreased, the resulting MR is reduced.

Considering this larger network scenario, we show the results related to the average NPE. The ε parameter (Section 5.5) is crucial in this regard: a very low value of 0.5 km if compared with the target area of each peer (10 GB*1.5 km) was set. Results confirm that on average the error is around ε for the duration of the simulation. The optimal choice for this parameter can be related to the requirements of the particular application. For example, there may be a need for high accuracy across a very large covered area, e.g. road/highway monitoring system (Figure 9).

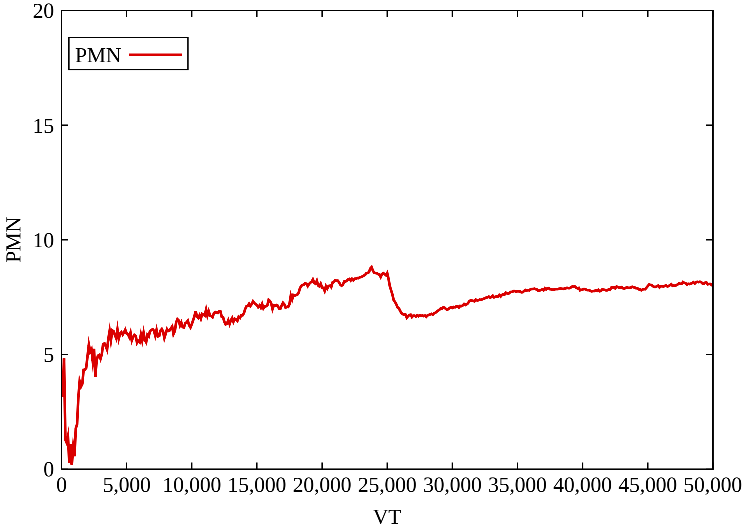

Figure 7.
PMN larger network

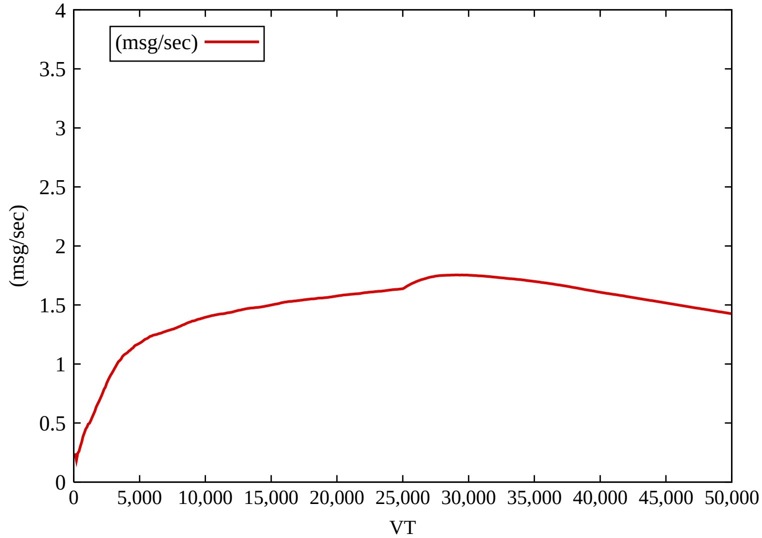

Figure 8.
MR larger network

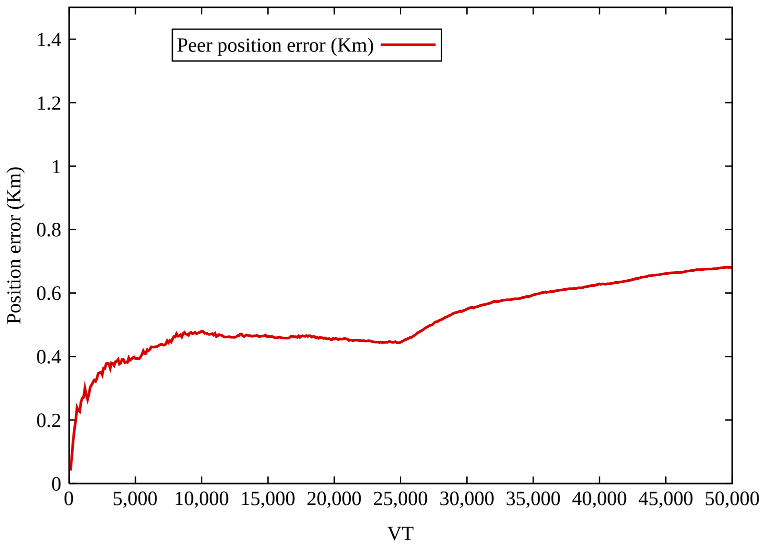

Figure 9.
NPE larger network

5.5 Evaluation of the position update mechanism

This scenario was created to assess the effects of ε variations on protocol performance. Using a network of 2,000 nodes (10 GB and a thickness of 1.5 km), we ran multiple simulations by varying ε between 0.1 and 1.35 km in 0.25 km steps. As previously stated, ε represents a displacement threshold, used in the position update procedure. A low value means that updates of peer positions are performed very often when users change their locations, whereas a high value causes infrequent updates.

The accuracy of information stored in GB is clearly related to the value of ε. Figure 10 shows the PMN with multiple ε values and we can see that there is a noticeable spread in the PMN results. This behavior is justified by the fact that a large ε value may lead to the erroneous exclusion or removal of a peer from the GB, resulting into accuracy loss and inconsistency.

The analysis of the NPE (Figure 11) shows that the average error is slightly larger than the threshold as there is an additional little variation introduced by peers' mobility and information's distribution among available nodes.

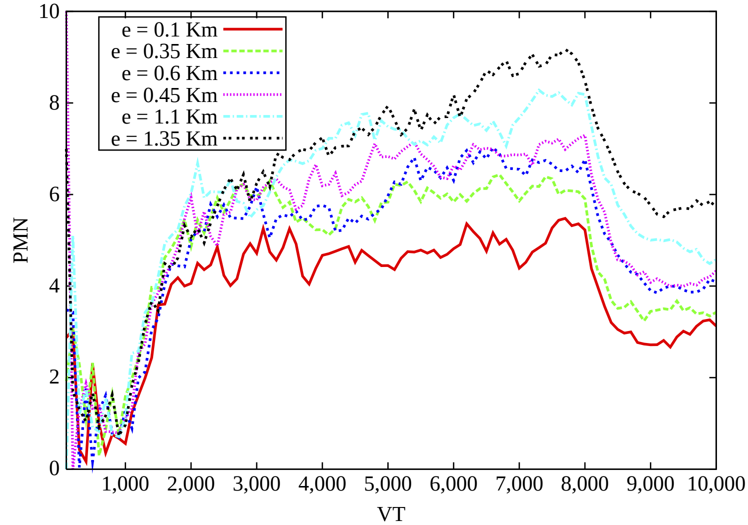

Figure 10.
PMN position update evaluation

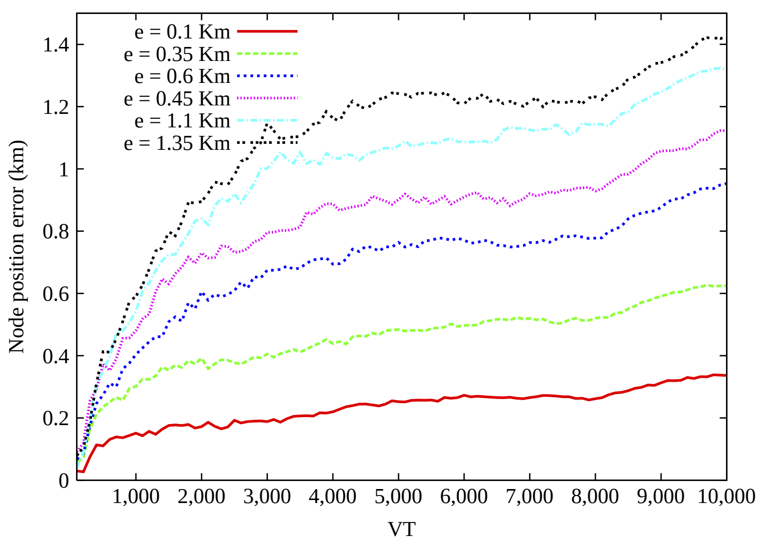

Figure 11.
NPE position update evaluation

Another important aspect related to these analysis is the number of exchanged messages. A small value of ε that results in a reduced error of position is strongly correlated with an increased value of MR. Figure 12 shows the results of different configurations and suggests that a value between 0.35 and 0.6 km can be a good compromise in terms of messages and accuracy for the chosen set of parameters.

This scenario is useful to understand the importance of the ε parameter and how we can make a better use of it. Clearly, this parameter is strongly related to application requirements, for which a careful analysis during the design phase gives the opportunity to reduce the number of exchanged messages without a great impact on global accuracy.

5.6 Robustness evaluation

A common element of all peer-to-peer systems that affects their performance is the high node dynamics due to churn. This section reports DGT results related to a very pessimistic scenario where the initial growth of the network is followed by a stabilization interval without new joins, and finally by a high churn phase. In the latter, a predefined portion of active peers (evenly distributed over the simulated area) disconnects at once to let us evaluate the overall robustness of the DGT system. Simulations are based on vehicular network mobility model using generated paths around Parma. The network is characterized by 1,000 active peers with the same growing behavior of previous described scenarios and using dynamic discovery period with a range of [1.5; 6]min depending on the number of new found nodes at the previous discovery iteration. The number of GB is five with a thickness of 0.5 km and $\varepsilon = 0.1$. A varying degree of node disconnection and the related PMN distribution have been analyzed. Figure 13 shows that the PMN for different fractions of disconnected peers maintains the same value (between 8 and 10 percent) without any significant variation. To understand the reason of such a good result, consider a peer that is aware of N neighbors. If M of the N neighbors leave the network unexpectedly, the peer may incur in false positives. Fortunately, the period during which the peer is not aware

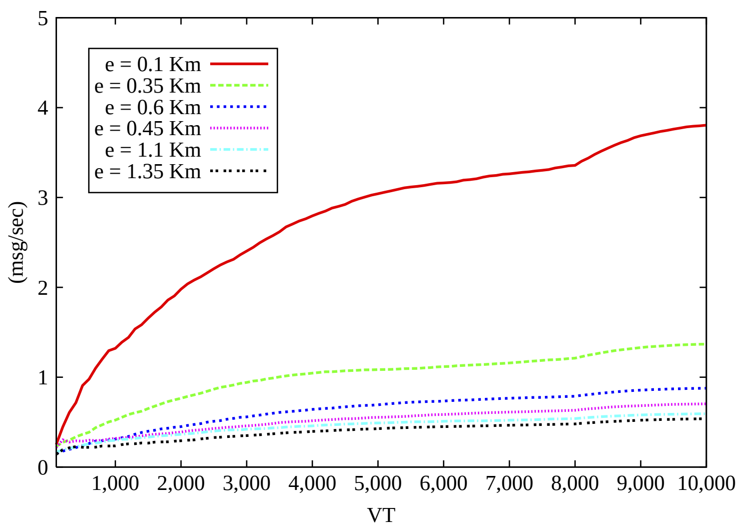

Figure 12.
MR position update evaluation

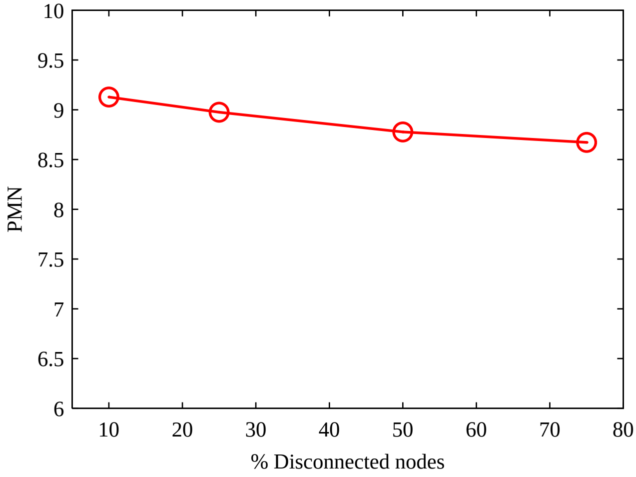

Figure 13. PMN for different percentages of disconnecting nodes

of the changes in its neighborhood is usually very short, because the peer sends maintenance messages, whose frequency increases with mobility. It is an interesting and important result that validates and confirms the robustness of the implemented DGT overlay which can efficiently manage abruptly and massive disconnections and consequently will handle at ease normal behaviors of active users in peer-to-peer networks. This results is also supported by the graph in Figure 14 that shows how the PMN is distributed in GBo revealing that is always very low and not significantly affected by peer disconnections.

5.7 Urban environment analysis

After the encouraging results shown in previous scenarios, the system behavior has been evaluated using the same mobility model of the previous section.

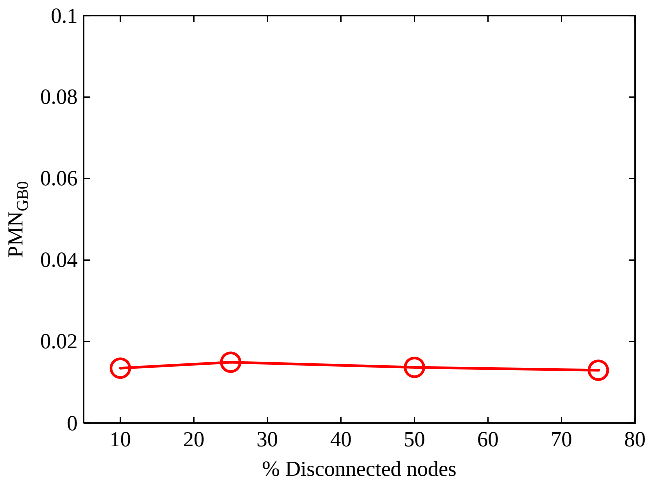

Figure 14. PMN_{GB0} for different percentages of disconnecting nodes

The analysis is divided in two different parts, with the first one focused to confirm previous results in a better modeled mobility scenario. The second part aims at evaluating how the size of the peer's local region of interest (as expressed by the number of GB K) affects DGT performance, that is the PMN, as also in this scenario, we are interested in finding all active nodes in the region of interest of a generic peer. Both simulation types refer to a square region surrounding the city of Parma, having a GB thickness of 0.5 km, $\varepsilon = 0.1$ as well as a dynamic discovery period with a range of [1.5; 6]min: as in previous analysis.

The first analysis considers a constant number of GB equal to five (covering a region of interest of $\cong 19\,km^2$) and monitors the variation of the overall PMN to different peer distribution in the network.

Simulation life is initially characterized by a growing number of active users that step by step join the network, start moving, exchanging messages, and discovering their neighbors. This phase is followed by a stable period without new joins or disconnections where the activities of the system proceed normally according to nodes movement and behaviors. Figure 15 shows the PMN value along all the simulation and confirms that the number of missing nodes is really low and around 5 percent for different sizes of the peer network. The second evaluation takes into account the effects of the variation of the number of GB (K) and the related covered area for a DGT node. Considering, as previously described, a reasonable high dynamic discovery period, it is possible to see in Figure 16 how the PMN value evolves according to the growth of K. The selected time interval allows to maintain a low PMN until the number of GB is equal to six (area $\cong 28\,km^2$) otherwise, the value grow very fast. This is of course, related to the discovery time because a larger area implies, on average, an increased number of nodes that change their position, and consequently requires the search procedure to be scheduled more frequently. To have a complete picture of the situation, it is very important to investigate how this PMN value is distributed among available GB and in particular in the first one that contains the knowledge about the closest

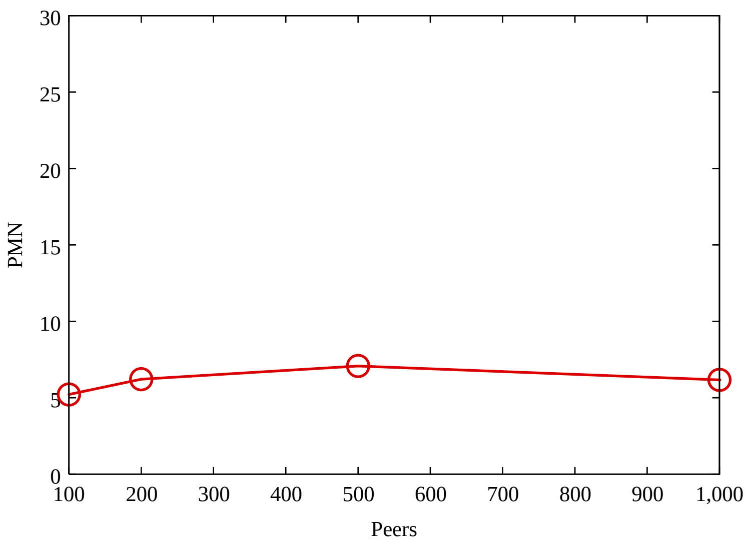

Figure 15.
PMN value for different network sizes

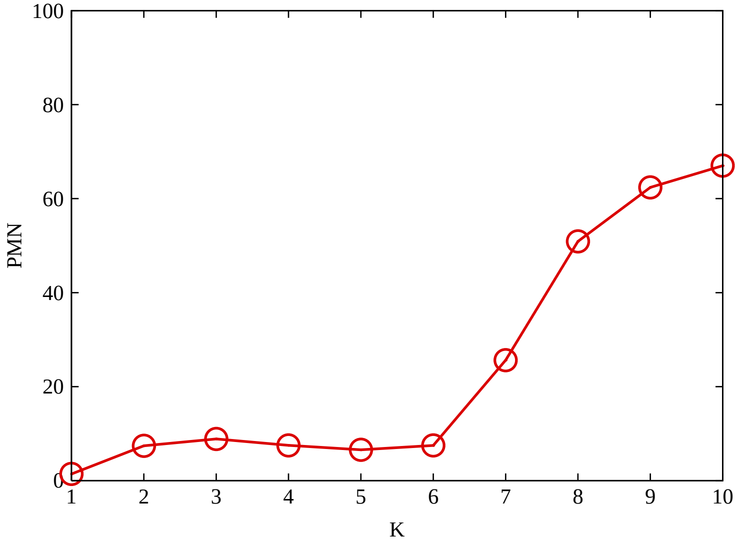

Figure 16.
PMN results for different *K*-values

neighborhood for a peer. Figure 17 shows also in this case, the good performance of the DGT approach that allows to keep the PMN near to zero for all tested GB configurations.

5.8 Discussion

Previously illustrated scenarios have shown that the effectiveness and efficiency of the protocol depend on the following system parameters:

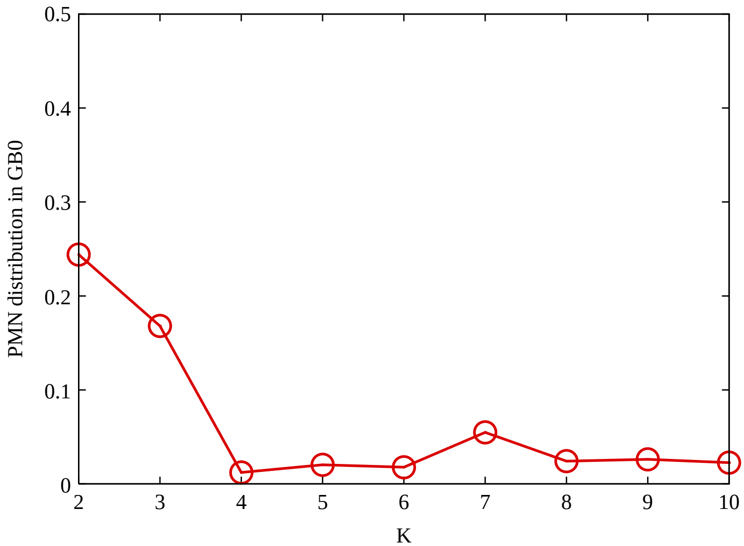

Figure 17.
PMN distribution in GB0 for different *K*-values

- Target area A (as covered by the GB).
- Discovery period (T_d).
- Position update threshold (ε).

How to configure optimally these parameters ultimately depends on the application. In the paper, we focused on a very "extreme" situation, where the objective of each peer was to discover all of its surrounding peers within a quite large area. As a matter of fact, some types of applications may require such a tight constraint, while others may have less stringent requirements.

An example application based on the DGT overlay may be a totally decentralized traffic information system that allows to efficiently disseminate information about urban traffic status such as accidents, jams, potholes or bad surface conditions. This application would not need that each peer knows all its neighbors. Potentially, a limited but well distributed knowledge would be sufficient to properly send messages and notifications to interested users (in terms of events and locations). Key parameters of such an application would be T_d and ε, and their values should be chosen to keep the neighborhood updated in highly dynamic scenarios, e.g. a vehicular network where location changes are very fast. At the same time, parameter A could be selected according to the distance that published localized information needs to cover.

A radically different application may be a Social Advertisement System, where shops and city offices would propose products and services to new or already registered users, according to their profile or to suggestions and feedbacks taken from their friend relationships. This system may be designed using two different instances of DGT protocols, one for nodes representing providers and another (completely different) for users. The former would catch all available peers in a target area A_p, which would become the key parameter of the architecture. The latter, instead, would build limited user lists, associated to a small area A_u used to route incoming messages and build awareness about the location of friends and services. In such a system, the frequency of the discovery procedure and the ε parameter may be set to obtain looser position updates, since for the sake of advertising misplacing users a few 100 meters does not make any practical difference.

As a final technical remark, we want to recall that the DGT-based peer neighborhood construction protocol presented in the paper uses a recursive strategy for routing messages inspired by Kademlia. Such an approach has proven to be effective, nevertheless alternative techniques for the implementation of peer discovery protocols may be considered. We are currently working on a new protocol to discover remote peers given the center of the circular region of interest. Such a protocol propagates queries within a triangular area whose height is the straight line that connects the location of the requestor to the remote point. With respect to the protocol illustrated in this paper, this one should be less bandwidth-consuming (propagating messages to a reduced number of peers). Probably, it will also be less comprehensive, when used to find nodes with specific features (e.g. owning a particular resource). Its performance evaluation will be presented in a future work.

6. Conclusions

Distributed localization for a massive number of users is a challenging problem with many useful applications. In this paper, we have introduced the concept of DGT referring to a decentralized system that allow to retrieve nodes or services located near any chosen geographic position using a distributed responsibility for maintaining information about the positions of active users. We presented also the particular instance of DGT, based on the peer-to-peer paradigm, that allow peers to localize all available nodes near to their geographic position enabling distributed and low cost application like city and traffic monitoring. We have shown that good performance can be achieved at low cost in terms of MR. All parameters in our protocol can be tuned in order to achieve the most suitable performance for the considered application. The distribution of missing nodes in available GB shows also that for the first container the percentage is almost zero and very low for the others GB except for the last one. These results allow us to envision many different applications for the DGT, such as vehicular networks, where the region of interest may range from the area surrounding the car, to retrieve specific information about accidents, or to any remote point of interest, to find information like the traffic status.

We also plan to investigate a formal model to support the general analysis of our DGT implementation. Moreover, we plan to take into account the estimation of peer trajectory (e.g. nodes traveling along highways) in order to reduce the number of exchanged messages. Finally, we intend to exploit local communication (Ad-Hoc networks) to directly retrieve available peers in the neighborhood and exchange useful information about GB data.

References

Aberer, K., Alima, L.O., Ghodsi, A., Girdzijauskas, S., Haridi, S. and Hauswirth, M. (2005), "The essence of P2P: a reference architecture for overlay networks", *Proceedings of the 5th IEEE International Conference on Peer-to-peer Computing* (*P2P2005*), *Konstanz, Germany, August 31-September 2*.

Amoretti, M. (2009), "A survey of peer-to-peer overlay schemes: effectiveness, efficiency and security", in Bussum (Ed.), *Recent Patents on Computer Science*, Vol. 2 No. 3, pp. 195-213, Ed. Bentham Science.

Amoretti, M., Agosti, M. and Zanichelli, F. (2009), "DEUS: a discrete event universal simulator", *Proceedings of the 2nd ICST/ACM International Conference on Simulation Tools and Techniques* (*SIMUTools 2009*), *Rome, Italy, March 2-6*.

Aringhieri, R., Damiani, E., de Capitani di Vimercati, S., Paraboschi, S. and Samarati, P. (2006), "Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems", *Journal of the American Society for Information Science and Technology*, Vol. 57 No. 4, pp. 528-37.

Asaduzzaman, S. and Bochmann, G.V. (2009), "GeoP2P: an adaptive and fault-tolerant peer-to-peer overlay for location based search, CoRR:abs/0903.3759", *Proceedings of the 35th International Conference on Very Large Databases* (*VLDB*), *Lyon, France, August 24*.

Balasubramanian, A., Levine, B.N. and Venkataramani, A. (2007), "DTN routing as a resource allocation problem", *Proceedings of the Special Interest Group on Data Communication* (*SIGCOMM'07*), *Kyoto, Japan, August 27-31*.

Balfe, S., Lakhani, A.D. and Paterson, K.G. (2005), "Trusted computing: providing security for peer-to-peer networks", *Proceedings of the 5th IEEE International Conference on Peer-to-peer Computing* (*P2P2005*), *Konstanz, Germany, August 31-September 2*.

Ekambaram, V. and Ramchandran, K. (2010), "Distributed high accuracy peer-to-peer localization in mobile multipath environments", *Proceedings of the IEEE Conference on Global Communications* (*GLOBECOM 2010*), *Miami, FL, December 6-10*.

Granovetter, M. (1973), "The strength of weak ties", *American Journal of Sociology*, Vol. 78 No. 6, pp. 1360-80.

Guttman, A. (1984), "R-trees: a dynamic index structure for spatial searching", *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data* (*SIGMOD84*), *Boston, MA, June 18-21*, pp. 47-57.

Harwood, A. and Tanin, E. (2003), "Hashing spatial content over peer-to-peer networks", *Proceedings of the Australian Telecommunications, Networks and Applications Conference* (*ATNAC*), *Melbourne, Victoria, December 8-10*, p. 5.

Kesar, C. (2003), "Experimental games for the design of reputation management system", *IBM Systems Journal*, Vol. 42 No. 3, pp. 498-506.

Kher, S., Somani, A.K. and Gupta, R. (2005), "Network selection using fuzzy logic", *Proceedings of the IEEE Second International Conference on Broadband Networks* (*BROADNETS 2005*), *Boston, MA, October 3-7*.

Kovacevic, A., Liebau, N. and Steinmetz, R. (2007), "Globase.KOM – a P2P overlay for fully retrievable location-based search", *Proceedings of the Seventh IEEE International Conference on Peer-to-peer Computing* (*P2P2007*), *Galway, Ireland, September 2-5*.

Li, M., Lee, W. and Sivasubramaniam, A. (2006), "DPTree: a balanced tree based indexing framework for peer-to-peer systems", *Proceedings of the 14th IEEE International Conference on Network Protocols* (*ICNP*), *Santa Barbara, CA, November 12-15*, pp. 12-21.

Liu, B., Lee, W.C. and Lee, D.L. (2005), "Supporting complex multi-dimensional queries in P2P systems", *Proceedings of the 25th International Conference on Distributed Computing Systems* (*ICDCS'05*), *Columbus, OH, June 6-10*.

Maymounkov, P. and Mazieres, D. (2002), "Kademlia: a peer-to-peer information system based on the XOR metric", *Proceedings of the First International Workshop on Peer-to-peer Systems* (*IPTPS*), *Cambridge, MA, March 7-8*.

Namand, B. and Sussman, A. (2006), "DiST: fully decentralized indexing for querying distributed multidimensional datasets", *Proceedings of the 20th International Parallel and Distributed Processing Symposium* (*IPDPS 2006*), *Rhodes Island, Greece, April 25-29*, p. 10.

Seskar, I., Marie, S., Holtzman, J. and Wasserman, J. (1992), "Rate of location area updates in cellular systems", *Proceedings of the 42nd IEEE Vehicular Technology Conference* (*VTC'92*), *Denver, CO, May 10-13*.

Tanin, E., Harwood, A. and Samet, H. (2007), "Using a distributed quadtree index in peer-to-peer networks", *The VLDB Journal*, Vol. 16 No. 2, pp. 165-78.

Tran, D.A. and Nguyen, T. (2008), "Hierarchical multidimensional search in peer-to-peer networks", *Computer Communications*, Vol. 31, pp. 346-57.

Xiaolei, Z., Chun-Fai, L., Cho-Li, W. and Francis, C.M.L. (2009), "Towards pervasive instant messaging and presence awareness", *International Journal of Pervasive Computing and Communications*, Vol. 5 No. 1, pp. 42-60.

About the authors

Marco Picone is a PhD student in Information Technologies in the Distributed Systems Group of the Department of Information Engineering of the Università degli Studi di Parma, Italy. His research activity focuses on distributed and peer-to-peer systems, with particular interest in solutions that involve mobile devices. Application fields include: neighbor position discovery in peer-to-peer networks, vehicle-to-vehicle and vehicle-to-infrastructure communications, P2P approach for inter-vehicular networks, vehicular networks simulation and mobility model, and mobile-based sensing systems. He is a main designer of the open-source middleware sip2peer for the implementation of P2P overlays. Marco Picone is the corresponding author and can be contacted at: picone@ce.unipr.it

Michele Amoretti is a Research Associate at the Department of Information Engineering of the Università degli Studi di Parma, Italy. His research focuses on the formal and simulative characterization of complex and autonomic distributed systems, in particular, those based on the peer-to-peer paradigm. He is a main designer and a programmer of the DEUS simulation environment.

Francesco Zanichelli received a Laurea Degree in Electronics Engineering from the University of Bologna and an Information Technology PhD from the University of Parma. Francesco Zanichelli is currently an Aggregate Professor at the Dipartimento di Ingegneria dell'Informazione of the Università degli Studi di Parma. His research interests are broadly related to distributed and peer-to-peer architectures as well as multimedia and mobile systems, and extend from fundamental issues through middleware up to services and applications.