

Chapter 5

Novel Distributed Algorithms for Intelligent Transportation Systems

5.1 Introduction

While hybrid communication schemes, i.e., combining V2V with V2I capabilities, would inherently provide the most effective and robust solution, we remark that purely infrastructure-based communication does not limit the application level of data dissemination and processing to a specific centralized architecture. In fact, a V2I infrastructure does not necessarily imply a centralized organization, which would inevitably lead to scalability issues—for example, to cope with the information requirements of thousands or millions of vehicles moving around in a large metropolitan area. While multiple distributed (e.g., hierarchical) subsystems can be deployed to achieve better scalability, a completely decentralized peer-to-peer (P2P) approach is more appealing. Initially exploited within V2V schemes [1], P2P approaches have been recently followed also for implementing decentralized TIS [2]. In fact, P2P strategies allow responsibility decentralization, as well as computational and communication load balancing, which can be beneficial for smartphone-based Vehicular Sensor Networks (VSNs) [3].

In the context of P2P TISs, we have implemented the D4V architecture, based on opportunistic mechanisms for the dissemination of data generated by vehicle sensors and drivers. D4V requires no dedicated hardware and leverages upon COTS and worldwide available devices (such as smartphones), rather than dedicated devices. To the best of our knowledge, D4V is the only TIS providing, at the same time, massive scalability (because of its P2P nature), deployability (because of the light hardware requirements), and message configurability. D4V is based on a P2P overlay scheme denoted as Distributed Geographical Table (DGT) [4, 5]—indeed, D4V stands for *DGT for VSNs*. The DGT overlay scheme represents a scalable and robust infrastructure for application-level services, and relies on the unification of the concepts of geographical and virtual neighborhoods [4]. In the following chapters, we first illustrate the DGT, with a detailed performance evaluation. Then, we present the D4V.

5.2 Distributed Geographic Table

A *structured* decentralized P2P overlay is characterized by a controlled overlay, shaped in a way that resources (or resource advertisements) are placed at appropriate locations [6]. Moreover, a globally consistent protocol ensures that any node can efficiently route a search to some peer that has the desired resource, even if the resource is extremely rare. Beyond basic routing correctness, two important topology constraints guaranteeing that (i) the maximum number of hops in any route (route length) is small, so that requests complete quickly, and (ii) the maximum number of neighbors of any node (maximum node degree) is small, so that maintenance overhead is not excessive. Of course, having shorter routes requires higher maximum degree.

The Distributed Geographic Table (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.

The DGT is different from others P2P-base localization systems, where geographic information is routed, stored and retrieved among nodes, that are organized according to a structured overlay scheme. The DGT idea is to build up the overlay directly taking in account the geographic position of node and information allowing to build a network where overlay neighbors are also geographic neighbors and no additional messages are needed to obtain the closest neighborhood of a peer (Fig. 5.1). In the following sections, we present the DGT approach, using the P2P system notation introduced by Aberer et al. [7].

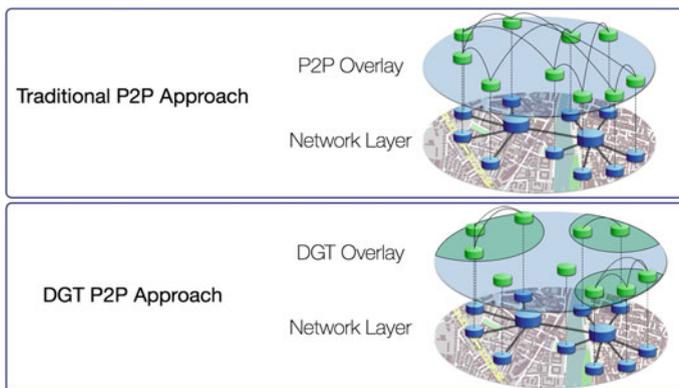


Fig. 5.1 Comparison between traditional P2P approaches and the distributed geographic table

5.3 Conceptual Framework

In a generic DGT overlay, the set of peers is called \mathcal{P} , each peer being characterized a unique $id \in \mathcal{I}$ (where \mathcal{I} is the space of identifiers). The association between a peer and an identifier is established by a function $F_p : \mathcal{P} \rightarrow \mathcal{I}$.

The space of world's coordinates is called \mathcal{W} and $w \in \mathcal{W}$, $w = \langle latitude, longitude \rangle$ is the generic location. Thus, a peer $p \in \mathcal{P}$ may be identified by the pair $\langle id_p, w_p \rangle$, where $id_p \in \mathcal{I}$ and $w_p \in \mathcal{W}$.

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):

$$d : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}. \tag{5.1}$$

The *neighborhood* of a geographic location is the group of nodes located inside a given region surrounding that location (as illustrated in Fig. 5.2). More precisely, given the set of all geographic regions delimited by a closed curve \mathcal{A} , the neighborhood is defined as

$$\mathcal{N} : \mathcal{W} \times \mathcal{A} \rightarrow 2^{\mathcal{P}} \tag{5.2}$$

where $2^{\mathcal{P}}$ 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 $A_t \in \mathcal{A}$ centered in t , let us define:

$$\mathcal{N} = \{p \in \mathcal{P} | w_p \in A_t\} \quad t \in \mathcal{W}, A_t \in \mathcal{A} \tag{5.3}$$

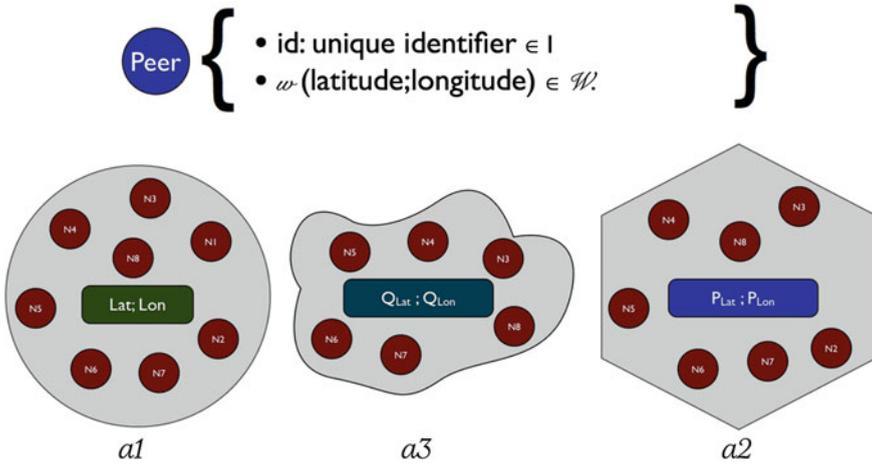


Fig. 5.2 The DGT routing strategy used to build and maintain a local or a remote neighborhood

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

$$C_t = \{w \in \mathcal{W} \mid d(w, t) \leq c_r\} \quad t \in \mathcal{W} \quad (5.4)$$

$$\mathcal{N} = \{p \in \mathcal{P} \mid w_p \in C_t\} \quad t \in \mathcal{W}, a \in \mathcal{A}. \quad (5.5)$$

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

5.3.1 Routing Strategy

The main service provided by the DGT overlay is the routing of requests for finding available peers in a specific area, i.e., to determine the neighborhood of a generic global position $w \in \mathcal{W}$.

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

$$\mathcal{R} : \mathcal{P} \times \mathcal{W} \times \mathcal{A} \rightarrow 2^{\mathcal{P}} \quad (5.6)$$

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 (local or remote). The idea, depicted in Fig. 5.3, is that each

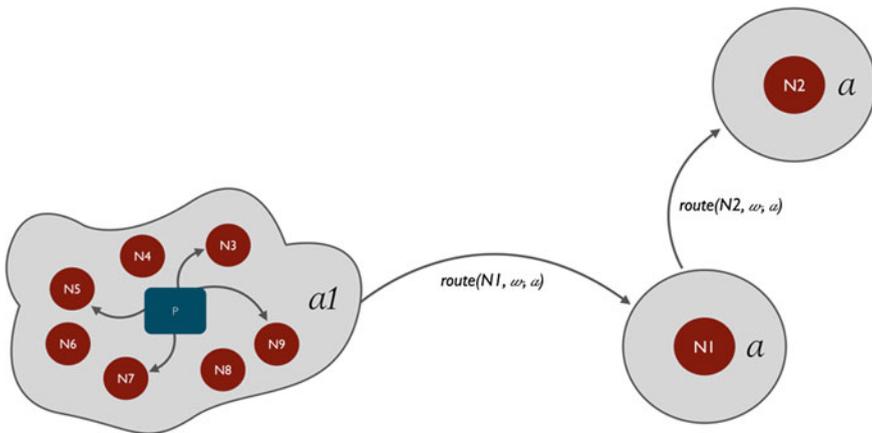


Fig. 5.3 The DGT routing strategy used to build and maintain a local or a remote neighborhood

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 accurate knowledge of nodes that are close to the peer and of a gradually reduced number of known nodes that will be used to forward long range geographic queries. This idea recalls Granovetter's theory of weak ties [8], stating that human society is formed by small complete graphs whose nodes are strongly connected (friends, colleagues, etc.). These clusters are weakly connected between each other, 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 human society an *egalitarian small world network*, i.e., a giant cluster with small separation degree and no hubs.

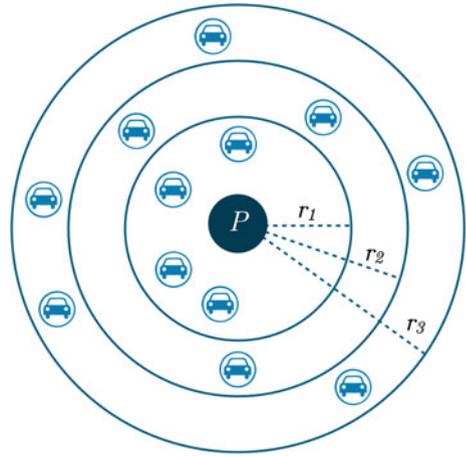
Whenever a single active node in the system wants to contact other peers in its area (e.g., to provide or search for a service), it does not need to route additional and specific discovery messages to its neighbors (or to a supernode responsible for a specific zone) in order to find peers that are geographically 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 [9], used, for example, in recent versions of the eMule client (as an alternative to the traditional eDonkey protocol) [10]. 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: in other words, each node in the network has lists containing up to k nodes, each list being associated to a given distance from the node 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.

Peer neighborhood construction in DGT uses the geographic metric, instead of Kademlia's XOR metric. Each node knows its *global position (GP)* retrieved with a 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 with respect to the node's position.

The main goal of the DGT 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 implement and provide 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.

Fig. 5.4 GeoBucket data structure



5.3.2 Data Structure

Every peer maintains a set of *GeoBuckets* (GBs), each one being a (regularly updated) list of known peers sorted by their distance from the *GP* of the peer itself (Fig. 5.4). GBs can be represented as K concentric circles, with increasing (application-specific) radii $\{R_i\}_{i=1}^K$ and thickness $\{r_i\}_{i=1}^K$, with $R_i = \sum_{j=1}^i r_j$. If there is a known node whose distance from the peer is larger than the radius of the outmost circle R_K , 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;
- *Global Position (GP)*—latitude and longitude retrieved with a GPS system or with other systems (e.g., GSM cell-based localization);
- *IP Address*—allowing to identify the node in Internet—if the peer is behind NAT, 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.

Moreover, each peer has a set of message types on which it is interested.

5.3.3 Network Join

The bootstrapping procedure is a common and crucial issue both in small and wide-spread P2P networks and several approaches have been studied and tested during last decade [11]. Essentially the basic bootstrapping process composed by two phases: find a remote peer, and connecting to it in order to receive a first group of nodes already active in the overlay. In a DGT overlay the idea is to use two different approaches (depicted in Fig. 5.5) according to the status of the nodes and the number

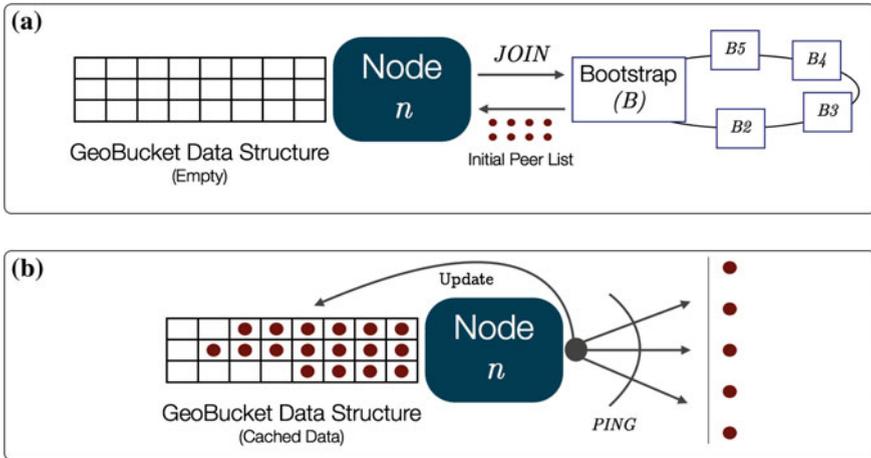


Fig. 5.5 Schematic representation of a single step of the DGT Lookup procedure

of known nodes or cached data. When a new node (First case in Fig. 5.5) wants to access the DGT overlay, it sends a join request, together with its GP , to a *bootstrap node*, that returns a list of 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 locations. 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. The second case is associated to a nodes that wants to re-join the DGT overlay after an offline period (case b in Fig. 5.5). In this situation it is highly reasonable that the node already has some cached data about its last GBs. In order to avoid the use of bootstrap node and, according to the delta between the current time the disconnection instant, the node could try to recover its distributed knowledge by pinging cached peer references and refresh its data structure. If after this refreshing procedure the reconnecting node is not able to use its information it will re-join using the bootstrap support cleaning its history.

5.3.4 Peer Lookup

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

Algorithm 1 Periodic Lookup Algorithm

```

1:  $i \leftarrow 0$ 
2: get  $\alpha$  nodes from geo-buckets (nearest to  $GP$ ):  $C_i = \{n_{1i}, \dots, n_{\alpha i}\}$ 
3: repeat
4:    $j \leftarrow 1$ 
5:   while  $j \leq \alpha$  do
6:     if  $n_{ji}$  not yet queried then
7:        $n_{ji}.FIND\_NODES(GP)$ 
8:     end if
9:      $j \leftarrow j + 1$ 
10:  end while
11:  get  $\alpha$  nodes (nearest to  $GP$ ) from the  $\alpha\beta$  results:  $C_{i+1}$ 
12:   $i \leftarrow i + 1$ 
13: until  $C_{i+1} == C_i$ 
14:  $f \leftarrow i$ 
15: get  $K$  nodes (nearest to  $GP$ ) from geo-buckets, not already in  $C_f$ 
16:  $j \leftarrow 1$ 
17: while  $j \leq K$  do
18:   if  $n_{ji}$  not yet queried then
19:      $n_{ji}.FIND\_NODES(GP)$ 
20:   end if
21:    $j \leftarrow j + 1$ 
22: end while

```

Node n searches in the GB associated to the requested GP. The final objective of the lookup (summarized in Algorithm 1 and schematically depicted in Fig. 5.6) is to find the $\alpha \leq K$ 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

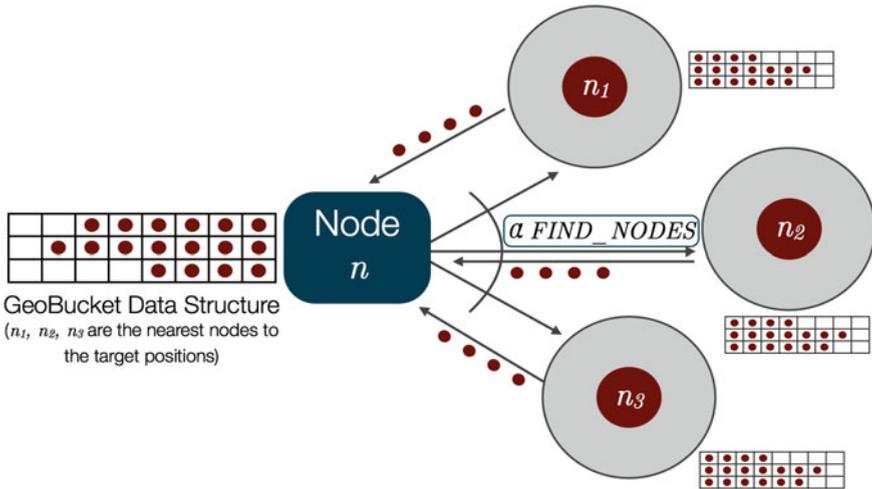


Fig. 5.6 Schematic representation of a single step of the DGT Lookup procedure

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 GBs. Such a peer set is denoted as $\mathcal{C}_i = \{n_{1i}, \dots, n_{\alpha i}\}_{i=1}^K$, where i is an integer index. The initiator sends parallel *FIND_NODES* requests, using its *GP* as target, to the α peers in \mathcal{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 *FIND_NODES* request (with the same target) to them. If a round of *FIND_NODES* fails to return a peer closer than the closest already known, the initiator re-sends the *FIND_NODES* to K closest nodes 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 *FIND_NODES*(*GP*) messages is in the worst case $f \cdot \alpha + K$, that depends on the spatial 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 to avoid the overlapping of the same type of operations.

The general idea is that soon after the bootstrap or when neighbor peers are highly dynamic, the period of the discovery process may be very small and may increase when the knowledge becomes sufficiently stable among active peers. We set a lower and an upper bound for the discovery period, i.e., respectively, T_{\min} and T_{\max} .

5.3.5 Position Update

Any active peer in the network can change its geographic position for many reasons (the user may be walking, driving, etc.).

To preserve the consistency of the DGT, each peer needs to periodically schedule a maintenance procedure that compensates network topology changes. The practical usability of a DGT critically depends on the messaging and computational overhead introduced by this maintenance procedure, whose features and frequency of execution are application-dependent.

When an active peer in the network changes its geographic position, it has to send updates of its *GP* to neighbors, in order to improve the accuracy of their knowledge. To avoid excessive bandwidth consumption, every peer communicates its position update to neighbors only if the displacement is higher than ε (km) (Fig. 5.8). If during this message exchange a peer receives a node's update confirming that the new position is out of its area of interest, the neighbor's reference is removed from the appropriate GB and a *REMOVE* message is sent to the peer (Fig. 5.7).

The DGT allows peers to have accurate knowledge of geographically close neighbors and a limited view of the outer world. However, whenever necessary, and with limited incremental computational and transmission costs, peers are able to find new connected nodes that are entering the target area. The described P2P localization scheme represents maybe the core layer of a vehicular network able to discover and inform drivers that are potentially interested in specific traffic messages or to data acquired by vehicle sensors.

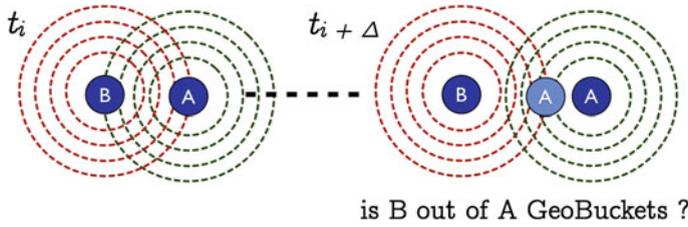


Fig. 5.7 DGT position update

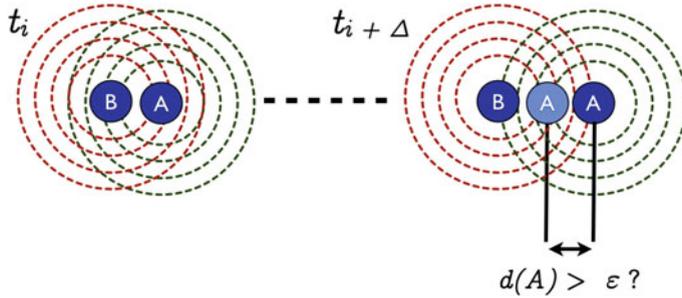


Fig. 5.8 DGT neighbor remove process

5.4 Analytical Model for Performance Evaluation

In this section, we first present an analytical performance evaluation framework of the DGT-based proactive neighbor localization algorithm. Furthermore, we carry out an extensive simulative analysis of our DGT implementation in several scenarios in order to evaluate algorithm performance, parameters influence and robustness.

The performance evaluation of our DGT-based peer neighborhood construction protocol will be carried using the following metrics:

- *PMN*: Percentage of Missing Nodes in the geo-buckets of a peer, with respect to those really present in the area.
- *MR* (msg/s): 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 GeoBucket and its actual position.

Assuming that N peers are distributed within a square surface with side of length L , the corresponding node spatial density, denoted as ρ , is N/L^2 . If nodes are static and uniformly distributed over the square surface, ρ is also the local node spatial density. In the presence of node mobility, the node distribution is likely to be non-uniform: the corresponding node spatial density can be heuristically estimated as $\delta N/L^2$, where $\delta \in \mathbb{R}^+$ is a compensation factor which takes into account the fact that the nodes could be locally denser ($\delta > 1$) or sparser ($\delta < 1$) than the average value ρ .

At a specific time, a peer wants to identify available geographic neighbors within a circular region of interest with radius r . This region, centered at the peer, is denoted as R and its area is $A = \pi R^2$. In general, within the region of interest of a node there are two classes of neighbors: detectable (i.e., nodes which can be detected by one or more nodes) and non-detectable (i.e., nodes which cannot be detected by any node).

Assuming that peers are distributed according to a two-dimensional Poisson distribution¹ with parameter ρ , the average number of nodes in the region R is $\overline{N}_{\text{tot}}^{(R)} = \rho \cdot A$. Let us denote by $x \in (0, 1)$ the percentage of non-detectable nodes in the region R (i.e., there are, on average, $x \cdot \overline{N}_{\text{tot}}^{(R)}$ non-detectable peers). Assuming further that the number of detectable peers in R has a Poisson distribution with parameter $\rho \cdot A \cdot (1-x)$, it follows that their average value is $\overline{N}_D^{(R)} = \rho \cdot A \cdot (1-x)$.

As described in Sect. 5.3, during each step of the discovery procedure a peer picks the closest α known neighbors (if available) and sends them simultaneous FIND_NODES requests centered in its geographic location. The goal of the interrogating peer is to retrieve detectable nodes in its area of interest. If, at the end of an iteration, no new node is retrieved, the discovery process ends and will be rescheduled according to a specific strategy.

In order to evaluate the number of discovered peers at each discovery iteration (without counting the same node more than once), the α FIND_NODES requests, scheduled at each discovery step, must be taken into account considering not only the single intersection between two peers but the multiple overlapped regions between the α contacted nodes. In Fig. 5.9, an illustrative scenario with $\alpha = 3$ overlapping circular areas is shown.

Since the intersection of α circular regions can be highly varying (depending on their relative positions), we simplify the analysis assuming that adjacent contacted peers are spaced by an angle $2\pi/\alpha$ and are positioned in the center of the corresponding radius of the circular region of interest of the reference peer. We denote as A_j the sum of the areas of the intersection region shared only by the requesting peer and j contacted peers. In Fig. 5.9, the areas $\{A_1, A_2, A_3\}$ are indicated. These areas will be computed using the Circles Intersection library for MATLAB². Explicit expressions (not shown here for the sake of conciseness) can be derived by means of the analytical method proposed by Fewell [12].

Under the above assumptions, the average number of new peers discovered after s steps can be written as

$$\overline{n}(s) = \begin{cases} 0 & s = 0 \\ l_0 & s = 1 \\ \overline{n}(s-1) + \sum_{j=1}^{\alpha} \overline{d}_j (\overline{n}(s-1)) & s \geq 2 \end{cases} \quad (5.7)$$

¹ This is an approximation. In fact, owing to node mobility, the local distribution is likely to be not Poisson. However, as we will consider only average values, the Poisson approximation will shown to be accurate.

² <http://www.mathworks.com/matlabcentral/fileexchange/5313>.

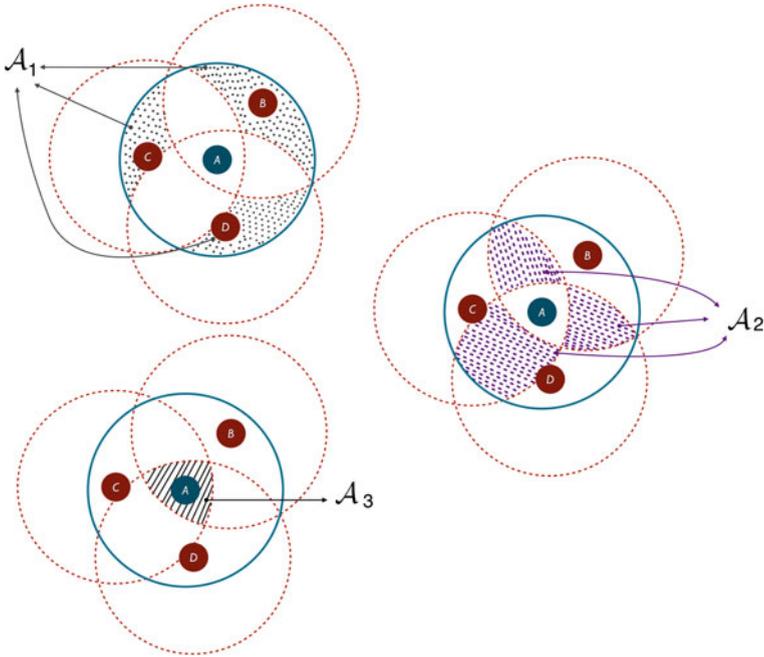


Fig. 5.9 Intersection regions (with corresponding areas $\{A_j\}$) between $\alpha = 3$ overlapping circular areas of interest

where l_0 is the initial size of the peer list; $\bar{n}(1)$ is the average number of initial peers (transferred to the peer of interest); $\bar{n}(s - 1)$ is the number of new peers discovered up to the $(s - 1)$ -th step ($s \geq 2$); \bar{d}_j represents the average number of new peers discovered in the region of area \mathcal{A}_j and can be expressed as follows:

$$\bar{d}_j (\bar{n}(s - 1)) = \rho \cdot A_j \cdot (1 - x) \cdot b_j (\bar{n}(s - 1)). \quad (5.8)$$

In (5.9), $b_j (\bar{n}(s - 1))$ is a heuristic function used to model the number of replicas obtained in the j -th intersection between the applicant's region of interest and the regions of interest of the queried peers. This parameter depends on (i) the number of nodes that share the same zone (i.e., j) and can answer with the same peer references and (ii) the the average number of known nodes at step $s - 1$ —in fact, the number of known nodes at each step needs to be taken into account to evaluate potential replicas. Taking into account the fact that if a node has knowledge of its neighbors, the probability of discovering an already known peer is higher, the following heuristic expression for b_j allows to derive accurate performance results:

$$b_j (\bar{n}(s - 1)) = \left[1 - \frac{\bar{n}(s - 1)}{N_D^{(R)}} \right]^j. \quad (5.9)$$

Finally, the average number of newly discovered nodes up to step s can be expressed as follows:

$$\bar{n}(s) = \bar{n}(s - 1) + \sum_{j=1}^{\alpha} \rho \cdot A_j \cdot (1 - x) \cdot \left[1 - \frac{\bar{n}(s - 1)}{N_D^{(R)}} \right]^j. \quad (5.10)$$

Note that the recursive analytical computation of $\{\bar{n}(s)\}$ stops when a pre-set peer discovery limiting number is reached.

In Fig. 5.10, the performance results predicted by the analytical model proposed above are compared with simulation results (obtained by means of the DEUS simulation platform, described in Appendix A), considering scenarios with (a) 500 peers

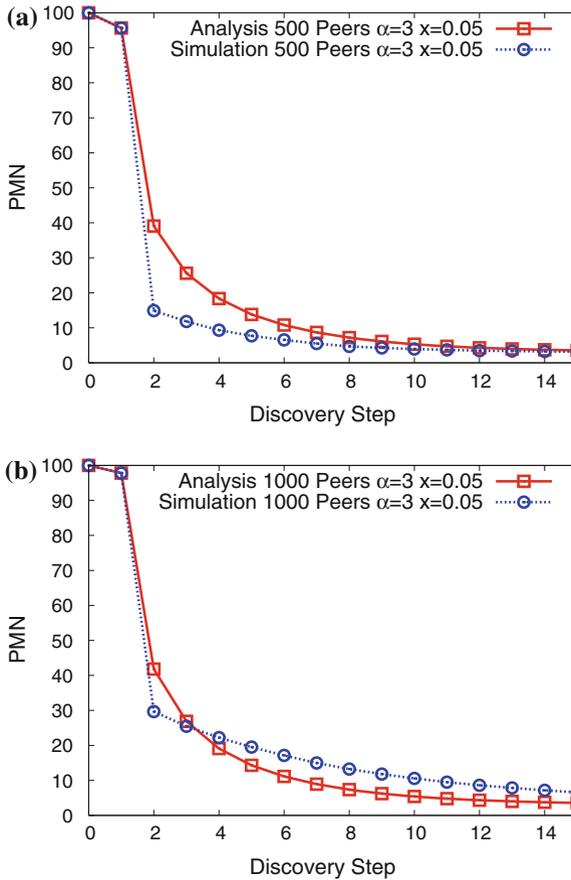


Fig. 5.10 PMN as a function of the discovery step, considering **a** 500 peers and **b** 1,000 peers. In all cases, $\alpha = 3$

and (b) 1,000 peers. In both cases (a) and (b), peers are distributed within a square surface with side of length $L = 6.53$ km, with an initial peer list size with $\bar{n}(1) = 10$ peers, a discovery limiting number of 100, and $x = 0.05$. It can be observed that analytical performance results are very close to simulation results, so that we can conclude that the accuracy of the analytical framework is satisfactory.

In order to investigate the impact of α , in Fig. 5.11 the PMN is shown, as a function of the discovery step, considering (a) $\alpha = 1$ and (b) $\alpha = 2$. In both cases, the number of active peers is set to 200. It can be observed that the agreement between simulations and analysis is even stronger than in Fig. 5.10. By observing the results in Figs. 5.10 and 5.11, it can be concluded that a small number of discovery steps is sufficient, regardless of the value of α , to significantly reduce the PMN.

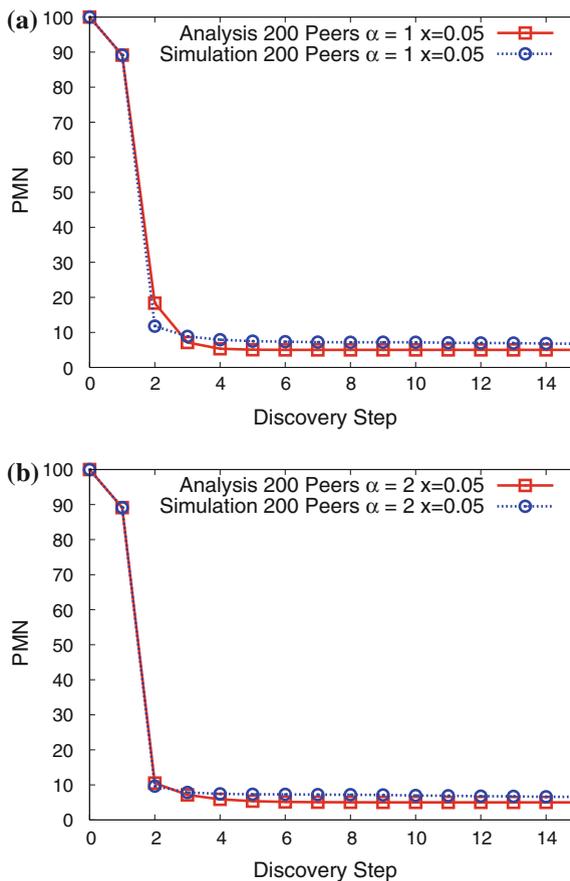


Fig. 5.11 PMN as a function of the discovery step, considering **a** $\alpha = 1$ and **b** $\alpha = 2$. In all cases, 200 peers

5.5 DGT and Mobility

Mobility models represent the movement of mobile users, and how their location, velocity and acceleration change over time. Such models are frequently used for simulation purposes when new communication or navigation techniques are investigated. Mobility management schemes for mobile communication systems make use of mobility models for predicting future user positions.

The mobility model is one of the fundamental elements in the performance evaluation of simulated network with mobile users such as V2V and V2I applications aiming at realistic mobility patterns.

We focus on the study of vehicular mobility models in order to evaluate the DGT approach in a dynamic scenario such as a Smart City. In this context multiple vehicles and user are moving at the same time querying about a location of interest and generating location based information or data such as traffic related messages or sensed data along the streets.

Figure 5.12 illustrates five major mobility model categories, which were defined by Hartenstein [13]:

- *Random Models:* Vehicular mobility is considered random and mobility parameters, such as speed, heading and destination are sampled from random processes. A very limited interaction between vehicles is considered in this category.
- *Flow Models:* Single and multi-line mobility models based on flow theory are considered from a microscopic or macroscopic point of view. The literature considers the following three different classes for flow models:

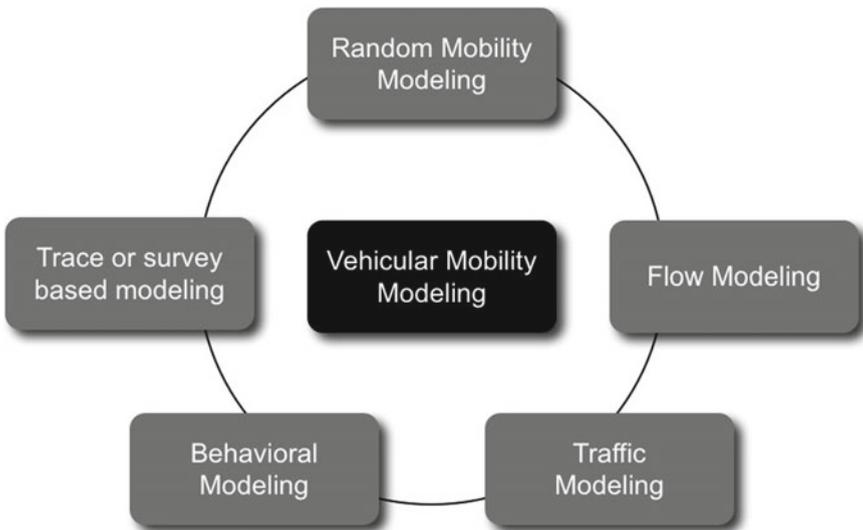


Fig. 5.12 Classification of vehicular mobility modeling approaches

- *Microscopic*: describe the mobility parameters of a specific car with respect to other cars in detail. Usually takes in account acceleration/deceleration, safe distance, reaction time or safe speed. Its high level of precision is reflected by high computational complexity.
 - *Macroscopic*: don't consider the mobility parameters of a specific car, but instead quantity of macroscopic meaning such as flow, speed or density are modeled
 - *Mesoscopic*: describes traffic flows at an intermediate level of detail. Individual parameters can be modeled yet of a macroscopic meaning. The aim is to benefit from the scalability of the macroscopic approach but still providing a detailed modeling close to microscopic models.
- *Traffic Models*: Trip and path models are described in this category, where each car has an individual trip or a path, or a flow of cars is assigned to trips or paths.
 - *Behavioral Models*: They are not based on predefined rules but instead dynamically adapt to a particular situation by mimicking human behaviors, such as social aspects, dynamic learning, or following AI concepts.
 - *Trace-Based Models*: Mobility traces may also be used in order to extract motion patterns and either create or calibrate models.

According to the concept map in Fig. 5.12, and to the ideas of Harri et al. [14, 15], mobility models intended to generate realistic vehicular motion patterns should include the following features (Fig. 5.13).

- *Accurate and realistic topological maps*: street topologies should manage different densities of roads, should contain multiple lanes, different categories of streets and associated speed limitations.
- *Obstacles*: obstacles should be intended as both constraints to car mobility and hurdles to wireless communications.

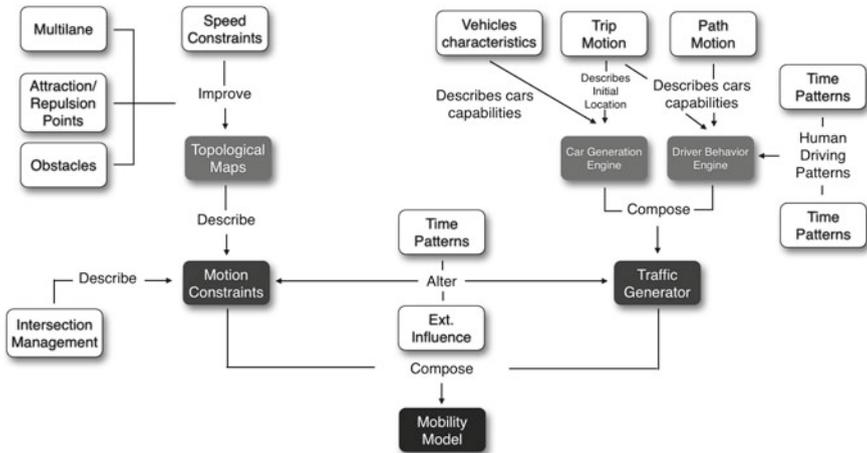


Fig. 5.13 Concept map of for the design of vehicular mobility models

- *Attraction/repulsion points*: initial and final destinations of road trips are not random. Most of the time, many drivers are driving toward similar final destinations or attraction points, or from similar initial locations or repulsion points, typically creating bottlenecks.
- *Vehicles characteristics*: each category of vehicle has its own characteristics, which has an impact on a set of traffic parameters. For example, macroscopically speaking, some urban streets and highways are prohibited to trucks depending on the time of the day. Microscopically speaking, acceleration, deceleration, and speed capabilities of cars and trucks are different. The accounting of these characteristics alters the traffic generator engine when modeling realistic vehicular motion.
- *Trip motion*: a trip is macroscopically seen as a set of source and destination points in the urban area. Different drivers may have diverse interests which affect their trip selection.
- *Path motion*: a path is macroscopically seen as the set of road segments taken by a car on its travel between an initial and a destination point. As in real life, drivers do not randomly choose the next heading when reaching an intersection as is the case in most vehicular networking traffic simulations. Instead, they choose their paths according to a set of constraints such as speed limitations, time of the day, road congestion, distance, and even the driver's own habits.
- *Smooth deceleration and acceleration*: vehicles do not abruptly break and move deceleration and acceleration models should be considered.
- *Human driving patterns*: drivers interact with their environments, not only with respect to static obstacles but also to dynamic obstacles, such as neighboring cars and pedestrians. Accordingly, the mobility model should control vehicles mutual interactions such as overtaking, traffic jams, or preferred paths.
- *Intersection management*: this corresponds to the process of controlling an intersection and may either be modeled as a static obstacle (stop signs), a conditional obstacle (yield sign), or a time dependent obstacle (traffic lights). It is a key part in this framework that however only has an influence on the motion constraint block, as the traffic generator block cannot not see the difference between a stop sign or high density traffic. Both are interpreted as a motion constraint.
- *Time patterns*: traffic density is not identical during the day. A heterogeneous traffic density is always observed at peak times, such as rush hours or during special events.
- *External influence*: some motion patterns cannot be proactively configured by vehicular mobility models as they are externally influenced. This category models the impact of accidents, temporary road works or real-time knowledge of the traffic status on the motion constraints and the traffic generator blocks. Communication systems are the primary source of information about these external influences.

Figure 5.14 and Table 5.1 illustrate the notation usually employed for the formal description of a vehicular mobility model where vehicle i will be considered as the reference vehicle. At time t , $x_i(t)$ and $v_i(t)$ represent respectively the position and speed of vehicle i . Indexes $i + 1$ and $i - 1$ represent the vehicle immediately in front and behind vehicle i with position $x_{i+1}(t)$ and $x_{i-1}(t)$, and with speed $v_{i+1}(t)$ and $v_{i-1}(t)$. We can additionally consider $\theta_i(t)$ as the heading of a vehicle i at time t .

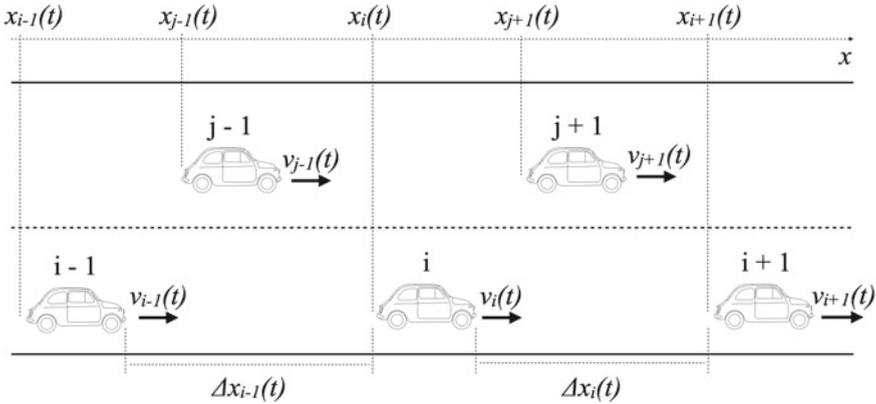


Fig. 5.14 Vehicular mobility model notation

Table 5.1 Vehicular mobility model description

Symbol	Definition	Dimension
Δt	Time step	s
a	Maximum acceleration	m/s^2
b	Maximum deceleration	m/s^2
L	Car length	m
v^{\min}	Minimum velocity	m/s
v^{\max}	Maximum velocity	m/s
v^{des}	Desired or targeted velocity	m/s
θ^{\max}	Maximum heading	rad
θ^{\min}	Minimum heading	rad
T	Safe time headway	s
Δx^{safe}	Safe distance headway	m
v^{safe}	Safe velocity	m/s
τ	Driver reaction time	s
μ	Stochastic parameter	[0;1]

Our model partially follows the approach of Zhou et al. [16] where the key idea is to use *switch stations* (SSs) connected via virtual tracks to model the dynamics of vehicle and group mobility. For example, our simulative analysis considers a square area around the city of Parma adding 20 SSs inside and outside the city district (see for example Fig. 5.15). Stations are connected to each other through virtual paths that have one lane for every direction, speed limitation associated with the street category and specific road density limit to model vehicle speed in traffic jam conditions. When a new car joins the network, it first associates with a random SS, then it selects a new destination station and starts moving on the connection path between them. This



Fig. 5.15 Example of simulated DGT-based VSN (in the city of Parma)

procedure is repeated every time the car reaches a new SS and has to decide its next destination.

Each switch station has an attraction/repulsion value that influences the user's choice for the next destination station. This value may be the same for each path in order to allow for random trip selection.

A set of parameters is associated with each car, thus affecting macroscopic and microscopic aspects of traffic circulation, like street and highway limitations (i.e., some types of vehicles are forbidden on particular paths) as well as acceleration, deceleration, and speed constraints.

We modeled different external events that may happen during the traffic simulation and alter drivers' behavior, such as accidents, temporary road works or bad conditions of road surface like ice, snow or potholes that can be detected by vehicle sensors.

Drivers not only interact with obstacles, but also adapt their behavior according to their knowledge about car surroundings. For example, they may try to change their path if they are informed about a traffic jam or an accident slowing or blocking, and they reduce their speed in proximity of locations characterized by bad surface conditions.

We consider microscopic flow modeling where mobility parameters of a specific car are described with respect to other cars. Several approaches take into account for example the presence of nearby vehicles when modeling the speed of the car

(e.g., FTM [17], Krauss [18], and IDM [19]). In particular, the FTM model has been implemented in our simulator because it is the most accurate for our scenario, with different speed limits for each virtual path, without high computational requirements. FTM describes speed as a monotonically decreasing function of vehicular density, forcing lower values when the traffic congestion reaches a critical point. In our case, the desired speed of a car moving along the points of a path p is computed according to the following equation:

$$v^{\text{des}} = \max \left\{ v_{\min}, v_{\max}^p \left(1 - \frac{k}{k_{\text{jam}}} \right) \right\} \quad (5.11)$$

where v_{\min} is the minimum car speed (depending on vehicle characteristics), v_{\max}^p is the speed limit related to the path, k is the current density of the road, given by n/l (n represents the number of cars on the road and l its length), and k_{jam} is the vehicular density for which a traffic jam is detected. As mentioned before, we also want to model the behavior of a driver in proximity of a road point with bad surface condition. The idea is that a conscientious driver, knowing that along his/her road there is a potential dangerous location, reduces the car speed according to the distance from that point. The safe speed v^{safe} is defined by the following equation:

$$\begin{aligned} v^{\text{safe}} &= \frac{d^2}{k_1} + k_2 \\ k_1 &= \frac{d_{\text{limit}}^2}{v^{\text{des}} - v_{\min}} \\ k_2 &= v_{\min} \end{aligned} \quad (5.12)$$

where d is the distance between the vehicle and path location with bad surface condition, d_{limit} is the limit from which the evaluation of safe speed starts, k_1 and k_2 are two constants depending on v^{des} and v_{\min} that are set to have the desired speed at limit distance and the minimum one near the dangerous location.

5.5.1 Mobility Model with Vertical Handover

Since we want to evaluate the robustness of our DGT-based localization algorithm in a dynamic urban scenario with mobile devices, considering several (possibly overlapping) regions characterized by different types of network coverage (see Fig. 5.16). To this purpose, we use one model to describe the mobility of vehicles, and another model for taking into account vertical handover.

Vertical handover or *vertical handoff* refers to a network node that change the type of connectivity it uses to access a supporting infrastructure, usually to support node mobility. For example, a laptop might be able to use both a high speed wireless LAN and a cellular technology for Internet access. Wireless LAN connections generally provide higher speeds, while cellular technologies generally provide more ubiquitous coverage. Thus the laptop user might want to use a wireless LAN connection whenever one is available, and to ‘fall over’ to a cellular connection when

Fig. 5.16 Example of area with different, partially overlapping network coverages

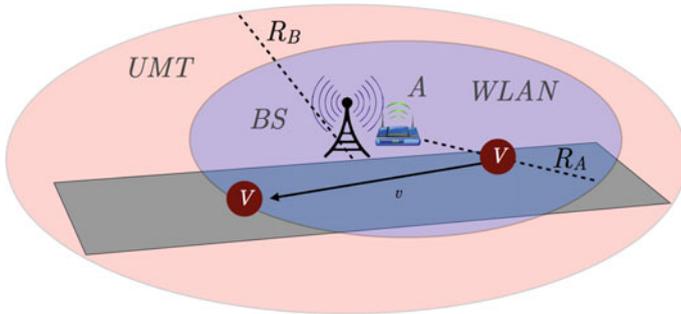
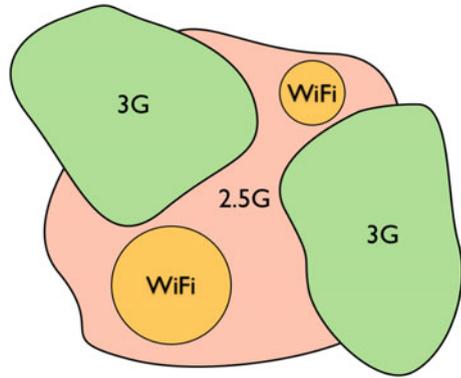


Fig. 5.17 Vertical handover scenario considering two overlapping regions (cellular base station and WiFi access point)

the wireless LAN is unavailable. Vertical handovers refer to the automatic fallover from one technology to another in order to maintain communication. This is different from a *horizontal handover* between different wireless access points that use the same technology in that a vertical handover involves changing the data link layer technology used to access the network.

Figure 5.17 shows a typical vertical handover scenario where a vehicle (or a generic mobile node) is moving with speed v under the coverage of two different network schematically represented as two circular overlapping regions respectively associated to a UMTS Base Station (BS) and to a WiFi Access Point (AP).

Regarding vertical handover, different algorithms for reducing the delay and the packet loss rate have been proposed. The Always Best Connected (ABC) concept was introduced by Gustafsson et al. [20, 21] to achieve seamless connectivity between WLAN and UMTS. The idea of using the vehicle speed as assessment criterion for vertical handover has been presented in [22, 23].

The vertical handover algorithm we adopt in our analysis is based on the approach presented by Esposito et al. [24], that bases the handover decision both on vehicle speed and handover latency. Our version of the model considers a vehicle V moving

with speed v^{des} in an environment characterized by several heterogeneous and overlapping access network at the same time. Defining SN (with bitrate B_{SN}) as the *servicing network* to which the user is connected, CN (with bitrate B_{CN}) the *candidate network* and L as the handover latency (the time interval during which the peer does not receive any data due to the socket switching), then the network switch is performed only if the time that the vehicle will spend in the area covered by the cell with higher bitrate (ΔT) is long enough to compensate for the data loss due to the switch overhead ($L < \Delta T$). The handover condition is defined as:

$$B_{CN} > \frac{B_{SN}}{1 - \frac{L}{\Delta T}} + \delta \quad (5.13)$$

where $\delta \in \mathbb{R}^+$ is an hysteresis factor used to avoid handover if the two competing networks have negligible bitrate difference. Since as previously described we are using a different mobility model and real vehicular city traces, instead of the Manhattan mobility model road composed of straight lanes used by the original authors, we need to redefine ΔT as follows:

$$\Delta T = \frac{\Delta x}{|v^{des}|} = \frac{R - d(V, CN)}{|v^{des}|} \quad (5.14)$$

where R is the radius candidate network station and $d(V, CN)$ is the geographic distance between the vehicle and the candidate network. The more the user is close to a cell site, the more he/she will stay within the coverage region of that cell site.

After each handover execution, the algorithm enters in idle mode for an inter-switch waiting period T_w , in order to avoid a high handover frequency that may happen when the vehicles travel on a border line between two different cells (ping-pong effects [25]).

The types of wireless connection we take into account are 2.5G and 3G mobile telephone technologies, as well as WiFi and WiMAX. In the first case we consider that connectivity is provided through horizontal handover where available cells located in the area allow the communication, and do not involve changing the technology used to access the network at the data link layer.

5.6 DGT Simulation

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. The interactions of the nodes determine their future state and that of the system [26]. Moreover, they usually exhibit high levels of concurrency and asynchrony, and their performance may be highly influenced by the changing environmental conditions of the environment, if they move.

For the qualitative and quantitative analysis of such systems, discrete event modeling and simulation (in which time jumps from event to event) are usually adopted [27]. In order to choose the proper simulation environment, the following criteria

should be taken into account: simulation architecture (the operation and the design of the simulator), usability (how easy the simulator is to learn and use), extensibility (the possibility to modify the standard behavior of the simulator in order to support specific protocols), configurability (how easily the simulator can be configured and with which level of detail), scalability (the ability to simulate how a P2P protocol scales with thousands, or more, nodes), statistics (how much the results are expressive and easy to manipulate), reusability (the possibility to use the simulation code to write the real application).

By looking at the state of the art, it is evident that almost every simulation tool targets a specific class of problems. Only few of them may be considered general-purpose. Among these, the most advanced, in our opinion, is CD++ [28], which is a modeling environment that allows to define and execute DEVS models [27]. OMNeT++ is another well-known general purpose discrete event simulation tool, which has been publicly available since 1997 [29]. Like CD++, also OMNeT++ is based on the concept of simple and compound modules. The user defines the structure of the model (the modules and their interconnection) using a topology description language called NED. OMNeT++ has been used in numerous domains from queuing network simulations to wireless and ad-hoc network simulations, from business process simulation to peer-to-peer network, optical switch and storage area network simulations.

Unfortunately, such tools are not particularly suitable for the simulation of distributed systems with thousands nodes, characterized by a high level of churn (node joins and departures), and reconfiguration of connections among nodes. To fill this gap, in 2009 we started a project for the development of an open source, Java-based, general-purpose discrete event simulation tool, called DEUS [30]. To simulate a distributed system at the application level, DEUS is particularly convenient, because of its extreme ease of use and flexibility. However, it does not provide packages for simulating networking layers, and we do not foresee to implement them. For this reason, until now the scheduling of application-level events to simulate the exchange of messages among nodes has been necessarily configured by the user, using reasonable values—which can be considered as a naive approach.

In order to implement realistic simulations of DGT-based systems, we integrated DEUS with Google Maps API. With the features provided by Google Maps API we created a simple HTML/Javascript control page which allows to monitor any simulated node, following it from starting to final position, and all the neighbors in its geo-buckets. This solution allows to study the protocol not only with specific P2P metrics—like message rate, miss ratio, number of peers—but also with a direct monitoring of peer behaviors during the simulation.

Recently, we introduced a general methodology to further improve our DEUS-based simulations, leveraging on a highly reliable and complete open source tool for the discrete event simulation of Internet systems, namely ns-3.³ The latter relies on high-quality contributions of the community to develop new models, debug or maintain existing ones, and share results. In Appendix A, we provide more details

³ <http://www.nsnam.org>.

about DEUS, and we describe our positive experience in integrating ns-3's LTE-EPC package [31] to support the network-aware simulation of the DGT. In next section we illustrate the packet delay model, after which we present the DEUS-based simulations of the DGT.

5.6.1 Packet Delay Model

To better characterize the communication among DGT peers in the urban environment, we defined the sub-model illustrated in Fig. 5.18, using ns-3 with the Lena LTE-EPC package⁴ [31]. The latter provides (1) the E-UTRA part of the Long Term Evolution (LTE) technology, dealing with physical and MAC layers, as well as Scheduler functionalities, and (2) support for the LTE RLC and PDCP protocol, together with EPC data plane features, such as the S1-U interface and the SGW and PGW entities. Shortly, such a ns-3 package supports the detailed simulation of end-to-end IP connectivity over LTE-EPC.

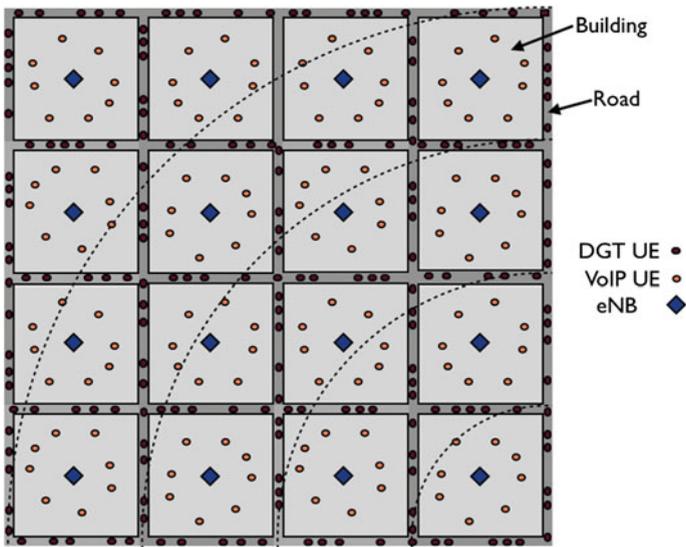


Fig. 5.18 Bird's-eye view of the simulated scenario, with $n = 200$ DGT nodes and $v = 96$ other UEs randomly placed within the buildings. The geo-buckets of the DGT node in the bottom right corner of the map are also drawn, to show that the side length of the considered area equals the GB radius

We considered DGT peers having GBs with radius of 2 km. Thus, we defined a square area having side length $l = 2$ km, with a grid of $r = 10$ roads (5 in N-S direction, and 5 in W-E direction) and vehicles running over them (with linear density

⁴ We used the version released the 23rd of January 2013.

Table 5.2 Spatial characterization of the buildings

Building #	Parameters	Values
1...16	x_{\min} [m]	100
	x_{\max} [m]	300
	y_{\min} [m]	100
	y_{\max} [m]	300
	z_{\min} [m]	0
	z_{\max} [m]	21
	# floors	7
	# walls type	ConcreteWithWindows

$\delta = 10$ vehicles/km). Drivers are provided with User Equipments (UEs), which execute a DGT-based application. The total amount of DGT UEs is $n = r\delta l = 200$.

Parallel roads are spaced by $l/4 = 0.5$ km. Between each pair of parallel roads, there are four large buildings with squared area, each one having seven floors. Table 5.2 describes such buildings in detail. Randomly located within each building, there are $v/16$ other UEs, where v is their total amount. The pathloss model is ns3::BuildingsPropagationLossModel.

On top of each building, exactly in the middle, there is an E-UTRAN Node B, also known as Evolved Node B, (abbreviated as eNodeB or eNB), i.e., a base station which serves a subset of the $n + v$ UEs. Table 5.3 reports the configuration parameters for the eNBs and the UEs. Regarding the eNBs, they have FDD paired spectrum, with 50 Resource Blocks (RBs) for the uplink, which means a nominal transmission rate of 50 Mbps, and 50 RBs for the downlink (50 Mbps)—like currently deployed LTE systems.

DGT UEs use UDP to send four types of DGT packets to each other. The first type, called **Descriptor**, is for neighborhood consistency maintenance purposes. Such a packet has the following structure:

- key: int (4 bytes)
- timestamp: float (4 bytes)
- lat: double (8 bytes)
- lng: double (8 bytes)

Table 5.3 LTE model: eNB and UE settings

Device type #	Parameters	Values
eNB	UIBandwidth [RB]	100
	DIBandwidth [RB]	50
	DIEarfcn	50
	UIEarfcn	18,100
	z [m]	23
	Tx Power	49
	Noise Figure	5
UE	Tx Power	23
	Noise Figure	9

where **key** is the identifier of the peer in the DGT overlay network, **timestamp** is the current time, and **lat/lng** indicate the location of the node. Considering also the 12 bytes header, the size of the DGT packet is 36 bytes.

The second type of packet is the **Lookup Request**, which is used to search for remote nodes placed around a specified location. Its structure is the following:

- **senderKey**: int (4 bytes)
- **lat**: double (8 bytes)
- **lng**: double (8 bytes)

where **senderKey** is the identifier of the peer in the DGT overlay network that sends the request, and **lat/lng** indicate the location of interest. With the 12 bytes header, the size of such a DGT packet is 32 bytes.

The third packet type is the **Lookup Response**, which is sent by a DGT node as a reply to a lookup request, if the node owns the searched resource/information. The structure of the packet is the following:

- **senderKey**: int (4 bytes)
- **lat**: double (8 bytes)
- **lng**: double (8 bytes)
- **descriptors**: Descriptor[20] (≤ 480 bytes)

where **senderKey** is the identifier of the peer in the DGT overlay network that sends the response, **lat/lng** indicate its location, and **descriptors** is a list of maximum 20 node descriptors. Considering the 12 bytes header, the maximum size of such a DGT packet is 512 bytes.

Finally, traffic information packets have the following structure:

- **trafficMessage**: String (30 bytes)
- **senderDescr**: Descriptor (24 bytes)
- **ttl**: float (4 bytes)
- **range**: double (8 bytes)

where **trafficMessage** is the message to be transmitted (e.g., “traffic jam”), **senderDescr** is the descriptor of the sender DGT node, **ttl** is the time to live of the message, i.e., the number of re-propagations it can be subject to, and **range** indicates the radius of the dissemination circle, which spatially limits the forwarding process.

We set an inter-packet interval of 50 ms for all types of DGT messages. Thus, the maximum rate is $512 \times 20 \simeq 10$ kB/s, while the minimum is $32 \times 20 = 0.64$ kB/s. In a dynamic DGT (the one we simulate with DEUS), packets are not sent periodically. For example, descriptors are sent only every ε meters. Lookup requests are sent only when necessary, as well as lookup responses. Traffic information messages are sent only when something interesting can be communicated to the other nodes (for example, a traffic jam or an incident).

The other UEs transmit and receive VoIP packets (using UDP) with a remote host located in the Internet. Such packets have a 12 bytes header and a 13 bytes payload, and inter-packet interval of 20 ms (we considered the AMR 4.75 kbps codec).

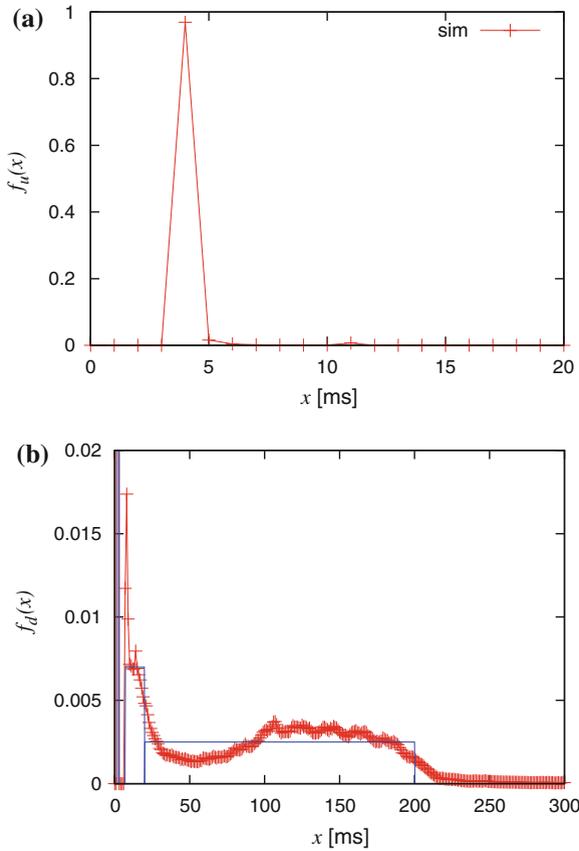


Fig. 5.19 PDFs of the uplink (a) and downlink (b) delays for DGT packets (for the case with $\delta = 5$ vehicles/km), obtained with ns-3

Each eNB has a scheduler which allocates RBs (which are the smallest elements of resource allocation) to users for predetermined amount of time. In these simulations, the Proportional Fair scheduler is used (ns3::PFFMacScheduler), which tries to maintain a balance between two competing interests: trying to maximize total wireless network throughput while at the same time allowing all users at least a minimal level of service.

As previously stated, every DGT UE sends a DGT packet (with randomly chosen type), with inter-packet interval of 50 ms. The ns-3 simulations were executed on a Ubuntu Linux 11.10 x86_64 machine with 16 GB of RAM and double quad core processor Intel(R) Xeon(R) Intel Xeon E5504 2.00 GHz. Each simulation was repeated with 20 different seeds for the random number generator.

For the DGT packet flow, we analyzed also the uplink and downlink delays—to this purpose, we modified the logger of the LTE LENA package in ns-3, to obtain a discretized probability density function (PDF) of the RLC packet delay. The PDF of

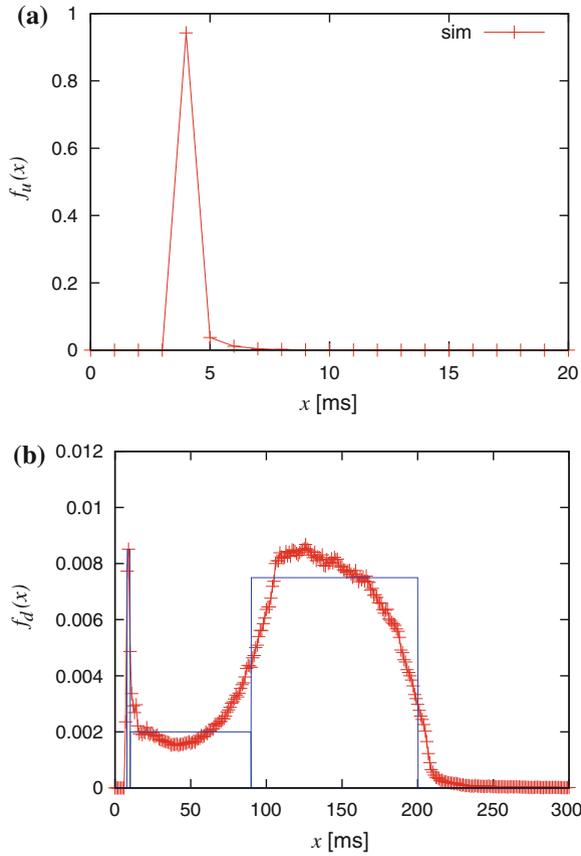


Fig. 5.20 PDFs of the uplink (a) and downlink (b) delays for DGT packets (for the case with $\delta = 10$ vehicles/km), obtained with ns-3

the uplink delay is basically a delta function, always centered in 4 ms, independently on the linear density of the vehicles (Figs. 5.19a, 5.20a, 5.21a). Instead, the PDF of the downlink delay is more complex, and is highly affected by the linear density of the vehicles (Figs. 5.19b, 5.20b, 5.21b). The PDF is then used to generate realistic packet delays in the DEUS-based simulations, using the well-known *inversion method* [32], which is based on the inverse probability theorem:

- choose the cumulative distribution function $F(x)$ of the random variable to be sampled;
- generate a set of uniform random numbers such that $R \sim U(0, 1)$;
- compute the random variate $X_i = F^{-1}(R_i)$.

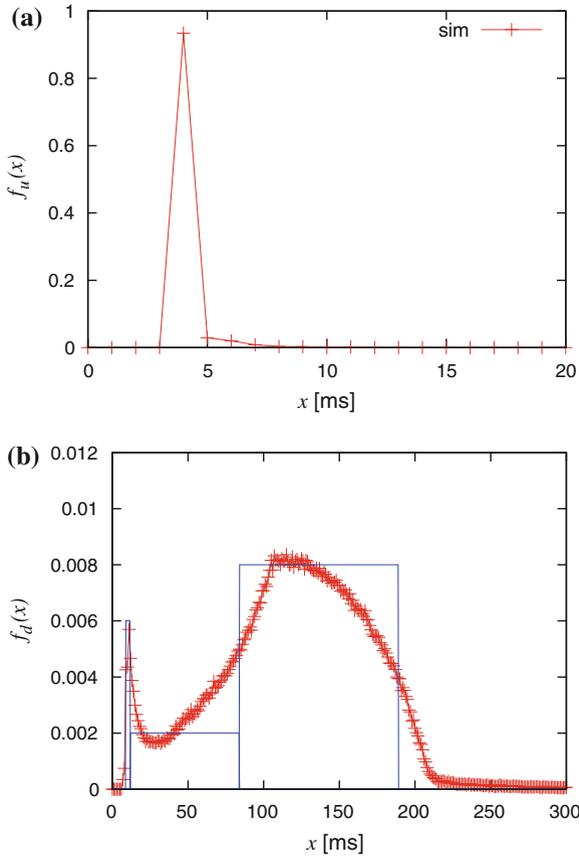


Fig. 5.21 PDFs of the uplink (a) and downlink (b) delays for DGT packets (for the case with $\delta = 20$ vehicles/km), obtained with ns-3

For practical implementation purposes, the discretized PDF of the downlink RLC packet delay is approximated by a piecewise constant function, whose numerical inversion is straightforward due to the reduced number of pieces (3).

5.6.2 DEUS Model

With the packet delay model described above, we obtained an improved DEUS simulation model of the DGT, with respect to the previous ones which used, for every transmission, an exponential delay with mean value obtained by considering the nominal uplink and downlink.

We simulated a DGT overlay with 1,000 mobile vehicles, over a period of 10 h. In the first half of such a period, the network grows from 0 to 1,000 nodes. In the

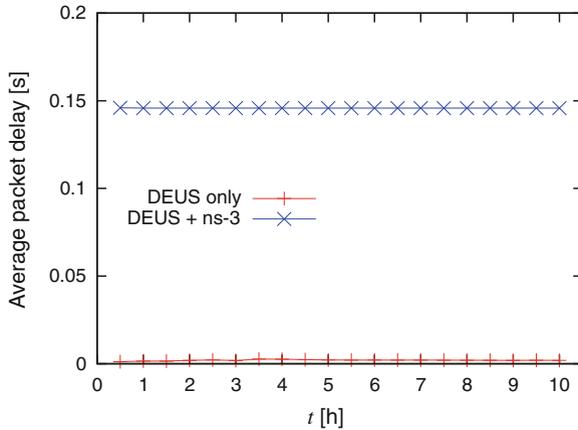


Fig. 5.22 Average packet delay, measured with DEUS, for the simulated DGT overlay network with 1,000 vehicles

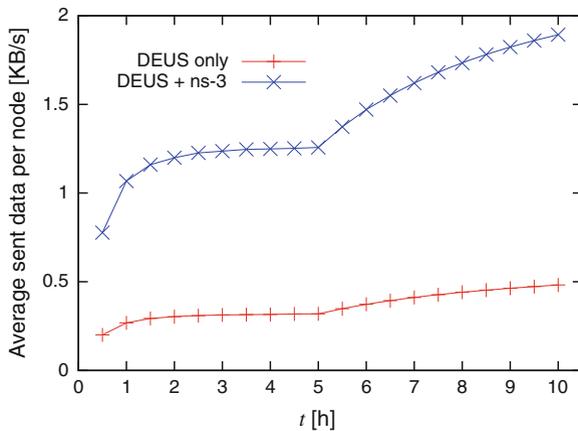


Fig. 5.23 Average amount of sent data per node, measured with DEUS, for the simulated DGT overlay network with 1,000 vehicles

second half, the size of the network remains stable. We logged the average packet delay and amount of sent data per node, computed on the whole overlay network. Figs. 5.22 and 5.23 compare the results obtained with the old simulation model, and those obtained with the refined one.

As we expected, in the refined model the average delay is higher than the one obtained with the naive model, which is based on nominal uplink and downlink values. Also the average amount of sent data is higher, because in the refined model we take into account also the header of the packets (12 bytes are 1/3 of Descriptor packets, which are the most frequently sent).

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 and is the FTM model presented in the previous section.

5.6.2.1 Evaluation of Geo-bucket Configuration

The first part of our simulation analysis aims at outlining how the choice of the geo-bucket configuration, in terms of number of GBs 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 10 h (corresponding to 10,000 virtual time units) while the second one has doubled size (2,000 peers) and time span (20 h, 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 5.4 presents all the considered cases.

Figure 5.24 shows the PMN for cases 1, 2 and 3, which refer to the first peer system (1,000 nodes over 10 h). We remark that the PMN remains moderate over all the simulated period, for all geo-bucket 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 %, while it decreases significantly when the peer network reaches a more stable state.

Table 5.4 Considered GB configurations

Case	#geo-buckets	GB thickness (km)	#Peer	Final VT
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

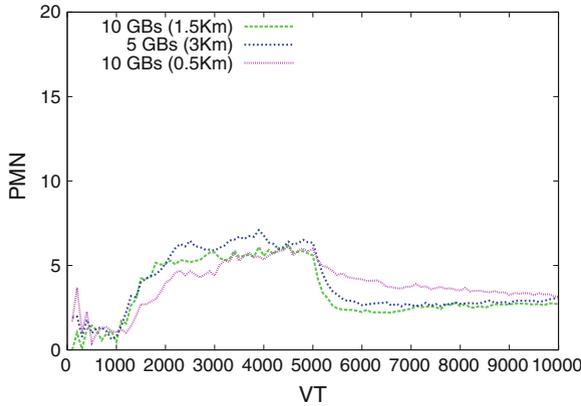


Fig. 5.24 PMN—GB configuration evaluation (cases 1, 2 and 3) with 1,000 nodes

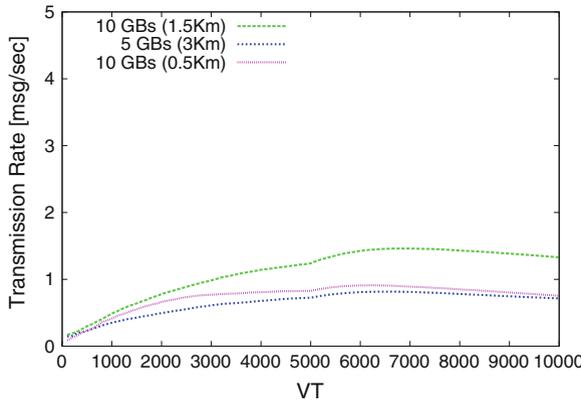


Fig. 5.25 Global MR—GB configuration evaluation with 1000 nodes

Another important metric that we must take in account to evaluate these different configurations is the MR. Figure 5.25 shows results for cases 1, 2 and 3. Given that a larger covered area ($\pi \cdot r_{GB}^2$) is potentially associated to a higher number of 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 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 geo-buckets (cases 4 and 5) in order to assess the protocol's behavior with a different distribution of nodes.

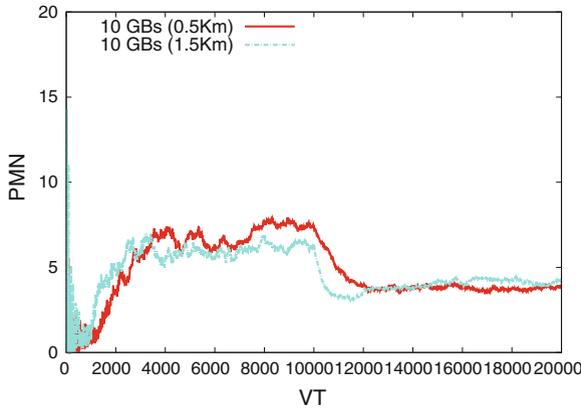


Fig. 5.26 PMN—GB configuration evaluation (cases 4 and 5) with 10 GBs and 2,000 nodes

Results in Fig. 5.26 show that also with a higher number of available peers and using two GB configurations the PMN is very small (under 10 and around 5 %). In the 4th case, 10 geo-buckets with 1.5 km thickness are used, which means a covered area of 706 km², whereas in case 5 we have only 5 geo-buckets with the same thickness for a covered area of 176 km². There is 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 (Fig. 5.25) that depends on peer density in the analyzed area.

We observe that the accuracy of the protocol shows little dependence on the configuration of geo-buckets (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 GBs in order to verify the knowledge evolution of active peers (Fig. 5.27).

Figure 5.28 is related to case 1, with 10 geo-buckets having a 1.5 km thickness. We already showed the associated PMN in Fig. 5.24 which stays very low for all the simulation at about 5 % or less. We analyze now how this value is split across different GBs. For the inner geo-bucket the percentage of missing node is around 0 % for the whole simulation's time. Predictably, the largest amount of missing peers is located in the external GBs that cover areas even very far from the peer. The outmost

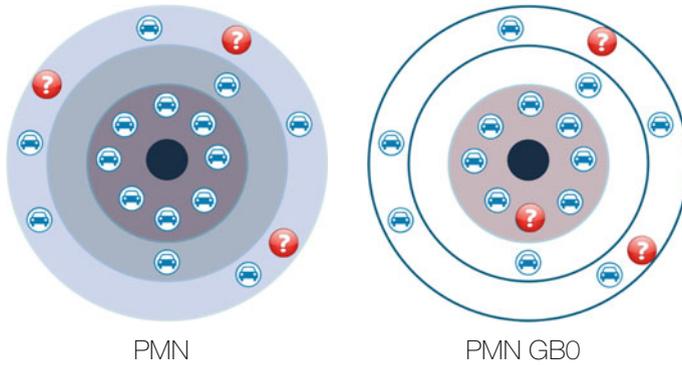


Fig. 5.27 Visual representation of the global PMN (for all GBs) and the same metric for a single GeoBucket (e.g., GB0)

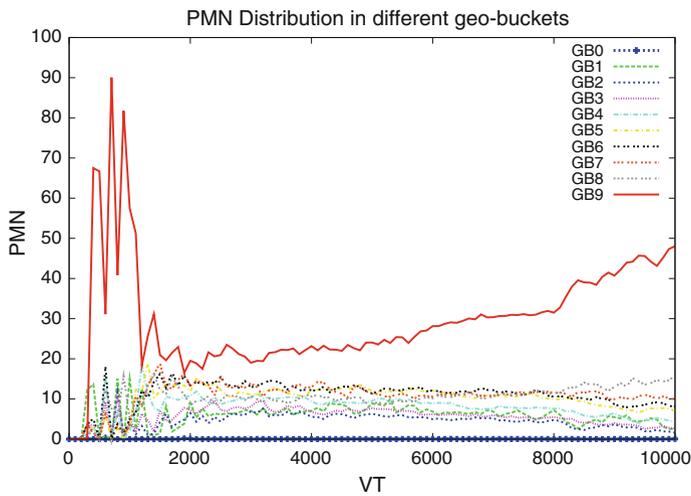


Fig. 5.28 PMN distribution in each geo-bucket—GB configuration evaluation (case 1)

GB, i.e., GB9 has the highest percentage of missing nodes and other geo-buckets limit the PMN under the 20 %. This is a very important result that shows how the protocol is very accurate and reliable and how it fulfills the DGT goal 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 example in vehicular networks, where it is very important to have the best knowledge of active users in a specific area of interest around the car.

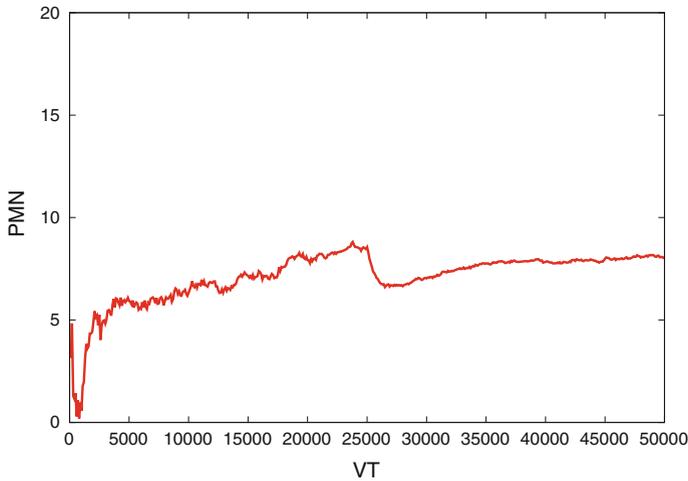


Fig. 5.29 PMN—larger network with $\approx 5,400$ peers

5.6.2.2 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 $\approx 5,400$ peers over a virtual time of 50 h using 10 geo-buckets with 1.5 km thickness.

Figure 5.29 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 %.

The cost in terms of exchanged messages (Fig. 5.30) is still very low, if we consider that the geo-bucket 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 node position error (NPE) (Fig. 5.31). The ε parameter is crucial in this regard: a very low value of 0.5 km—if compared with the target area of each peer ($10 \text{ GB} \times 1.5 \text{ km}$)—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.

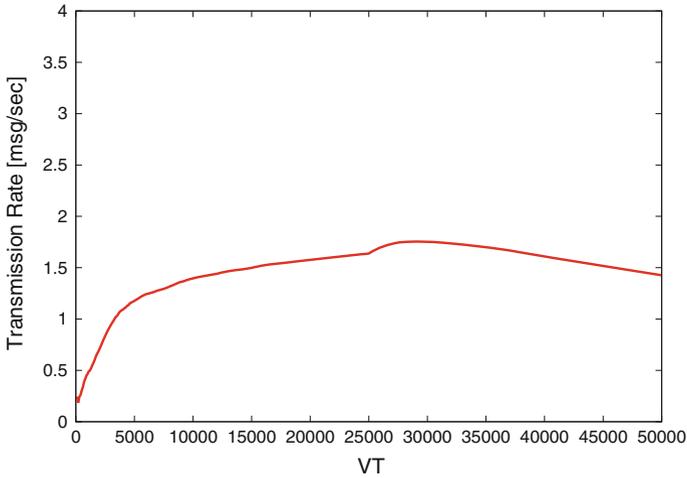


Fig. 5.30 MR—larger network with $\approx 5,400$ peers

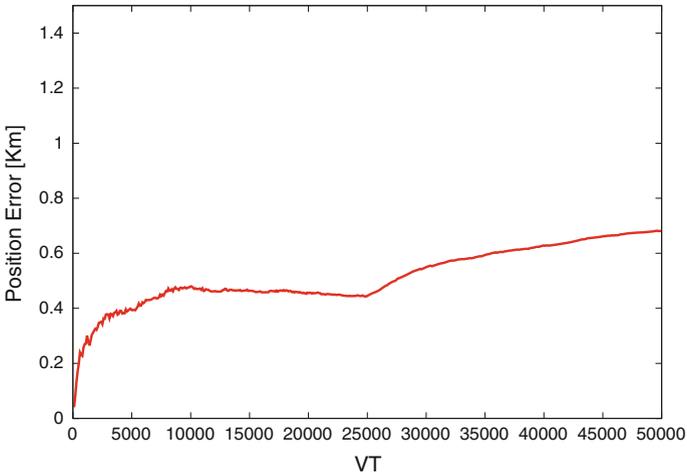


Fig. 5.31 NPE—larger network with $\approx 5,400$ peers

5.6.2.3 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 GBs 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.

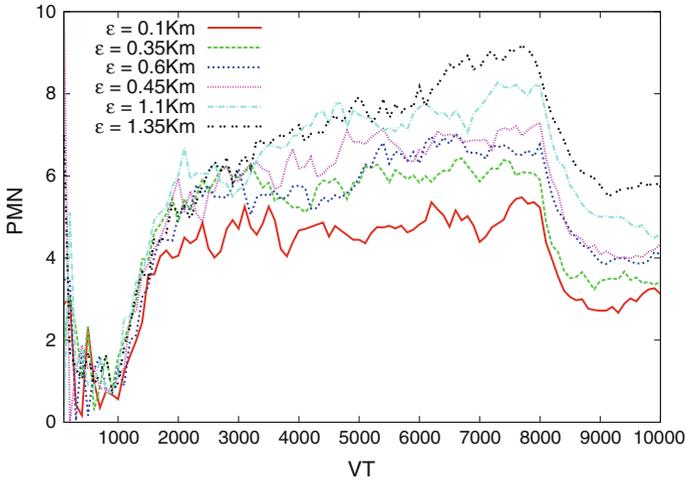


Fig. 5.32 PMN—position update evaluation for several values of ϵ

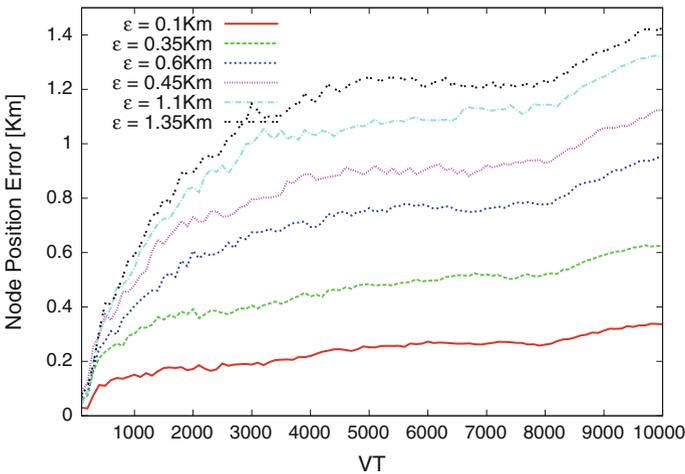


Fig. 5.33 NPE—position update evaluation for several values of ϵ

The accuracy of information stored in GBs is clearly related to the value of ϵ . Fig. 5.32 shows the percentage of missing nodes 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 GBs, resulting into accuracy loss and inconsistency.

The analysis of the NPE (Fig. 5.33) 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.

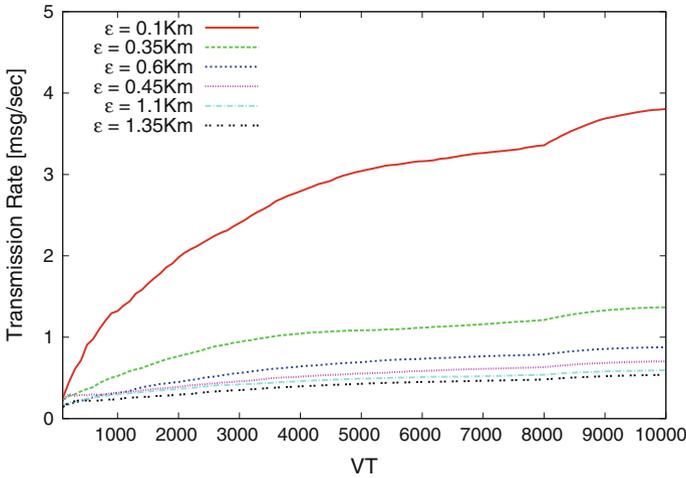


Fig. 5.34 MR—position update evaluation for several values of ϵ

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 5.34 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.

5.6.2.4 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. Simulation are based on a vehicular mobility model, with paths located in the city of 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 geo-bucket is 5 with a thickness of 0.5 km and $\epsilon = 0.1$. A varying degree of node disconnection and the related PMN distribution

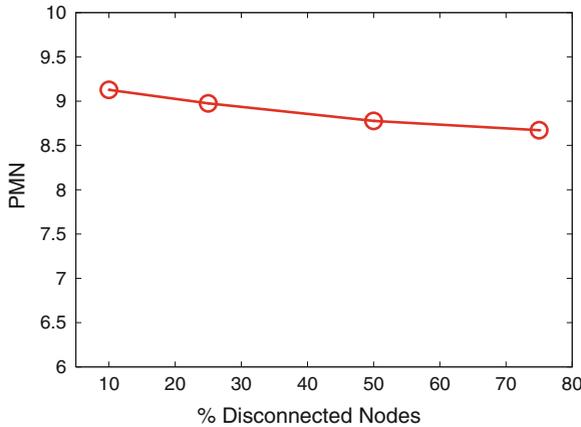


Fig. 5.35 *PMN* for different percentages of disconnecting nodes

have been analyzed. Figure 5.35 shows that the percentage of missing nodes for different fractions of disconnected peers maintains the same value (between 8 and 10 %) 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 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 P2P networks. This results is also supported by the graph in Fig. 5.36 that illustrates how the *PMN* is distributed in GB_0 revealing that is always very low and not significantly affected by peer disconnections.

5.6.2.5 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.

The analysis is divided in two different parts, with the first one focused to confirm previous results in a better modelled 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 geo-buckets— 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.

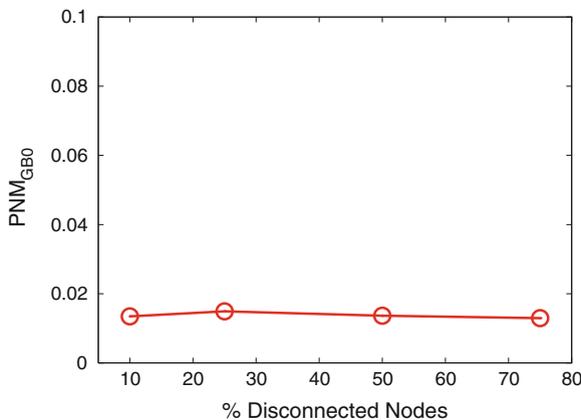


Fig. 5.36 PMN_{GB0} for different percentages of disconnecting nodes

The first analysis considers a constant number of geo-buckets equal to 5 (covering a region of interest of $\cong 19 \text{ km}^2$) and monitors the variation of the overall percentage of missing nodes to different peer distribution in the network.

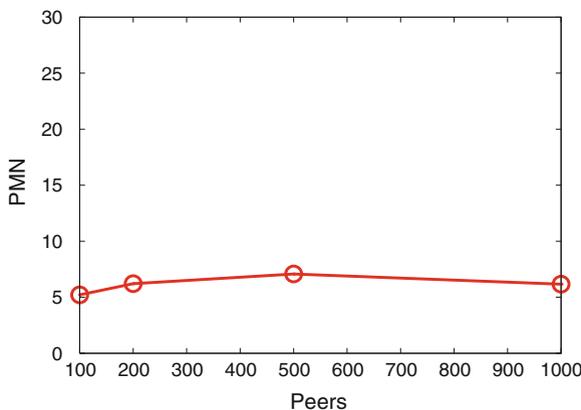


Fig. 5.37 PMN value for different network sizes

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 5.37 reports the PMN value along all the simulation and confirms that the number of missing nodes is really low and around 5 % for different sizes of the peer network. The second evaluation takes into account the effects

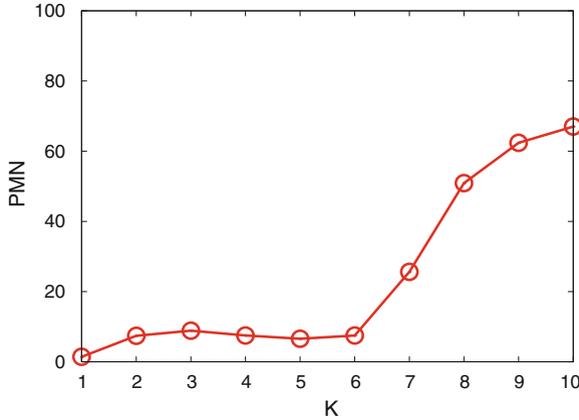


Fig. 5.38 *PMN* results for different K values

of the variation of the number of GBs (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 Fig. 5.38 how the *PMN* value evolves according to the growth of K . The selected time interval allows to maintain a low percentage of missing nodes until the number of GBs is equal to 6 (area $\cong 28 \text{ km}^2$) 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 GBs and in particular in the first one that contains the knowledge about the closest neighborhood for a peer. Figure 5.39 confirms also in this case the good performance of the DGT approach that allows to keep the PMN_{GB_0} near to zero for all tested GBs configurations.

5.6.2.6 Vertical Handover Analysis

The analysis of the robustness of DGT-based localization, considering the vertical handover model defined in previous section has been performed considering the same region of 10 km^2 -squared area centered on the city of Parma and the mobility model previously explained.

We have considered a DGT overlay where available nodes have $K = 3$ different GeoBuckets, with a thickness of 1Km and a dynamic discovery period ranging from 1.5 to 6 min, depending on the number of discovered nodes (if the latter decreases, then the period increases). Simulations cover ten hours of system life (10,000 virtual time units) and have been averaged over several execution runs with different seeds. Additional parameter values are $T_w = 20$ (s) to reduce pingpong effects and

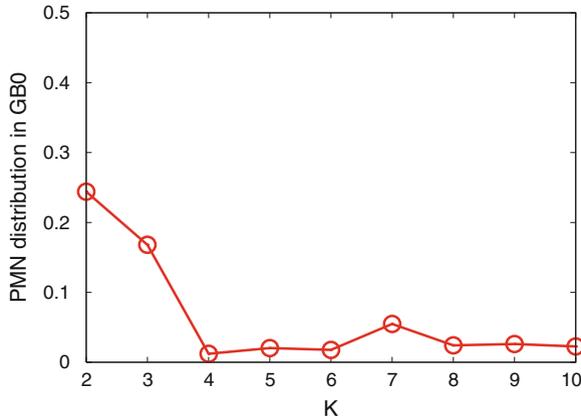


Fig. 5.39 *PMN* distribution in GBO for different K values

$\delta = 1$ (Mbps) as hysteresis value to avoid handover if the two competing networks have negligible data rate difference. Road accident events are scheduled during the simulation according to a Poisson process with mean inter-arrival value of 1,000 VTs. These and other events are sensed by vehicles and disseminated over the DGT through different message types.

In the evaluation, we have considered the following additional performance metrics:

- *RT(x)* [dimension: (s)]: Reconnection Time, i.e., the average time required by a temporary disconnected peer to recover the knowledge of its neighborhood and minimize the *PMN* value under x %.
- *Packet Loss/min*: average number of packets that fail to reach the destination per minute per peer. It takes in account both DGT and content dissemination packets.
- *% Coverage*: Estimated coverage percentage of traffic information messages at a certain time of the simulation. It is evaluated as the number of peers that actually received a specific message over those that should have it.

With the aim of improving the accuracy and the realism of simulated models, we performed multiple field measurements using Android smartphones (HTC Desire and Samsung Galaxy) on a vehicle moving along several Parma streets. In this way, we obtained experimental data about:

- *Uplink and Downlink Rates*: Real-world communication rates for uplink and downlink channels.
- *Cell Tower Information*: Information about cell towers located in the area of interest, such as geographic location, provider, connection type, measured distance to cell tower and RSSI value.

All measurements have been carried out with different smartphones and SIM modules of three italian providers (TIM, 3 ITA, Vodafone). Tower locations were

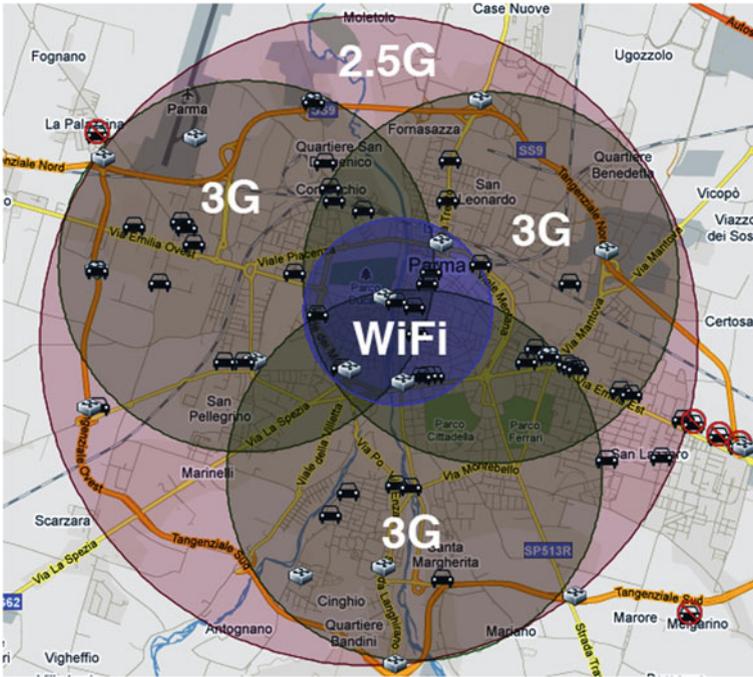


Fig. 5.40 Simulated network regions in Parma urban area

Table 5.5 Types and performance of the available network regions in the simulated urban area

#	Type	Uplink min-max (kbit/s)	Downlink min-max (kbit/s)
1	2.5G	30–90	60–170
2	3G	35–1,150	91–2,650
3	WiFi	100–2,000	2,000–10,000

used to build a map of available cell towers (the overall coverage is schematically shown in Fig. 5.40), each one characterized by a specific connectivity type and a coverage area with a 1.5 km radius. A WiFi region with a radius of 2.0 km is also located in the city center (which likewise approximates the coverage that is actually available in Parma center) to provide higher data rates where the density of peers is very high and consequently larger messages are exchanged among nodes to share the information about neighborhood. For each type of connectivity Table 5.5 reports the ranges of data rates experimentally obtained on the field and thus the limits for the values considered by the simulation.

A first simulative analysis has been carried out to evaluate the robustness of the DGT overlay with respect to vertical handover in five scenarios with different network coverage, considering the vertical handover latencies reported in Table 5.6. Simulated

Table 5.6 Vertical handover latency timetable

	Disc. (s)	2.5G (s)	3G (s)	WiFi (s)
Disc.	0	1	1	5
2.5G	1	0	0	5
3G	1	1	0	5
WiFi	5	5	5	0

Table 5.7 Simulated scenarios with different connectivity coverage of the urban area

Scenario	2.5G regions (%)	3G regions (%)	WiFi regions (%)
1	100	100	100
2	100	75	0
3	100	50	0
4	100	25	0
5	100	0	0

coverage distributions are summarized in Table 5.7, starting from a fully operational Scenario 1 and proceeding with a progressive decrease of available connection types for WiFi and 3G networks.

Graph (c) in Fig. 5.41 reveals how significantly the presence of a WiFi region in the first scenario and the related expensive vertical handover effect influences the reconnection time needed by a peer to recover the PNM under the 10 % of missing nodes. In scenarios where there is no WiFi area the required period is quite smaller. This behavior affects only marginally the number of packet lost per minute (b) and the global PMN distribution (a)—the latter results slightly higher in the first scenario compared to the others. Furthermore, the percentage of missing node in the inner GeoBucket (b) remains really low in all simulated scenarios, allowing for a high coverage of traffic information disseminated among nodes. Those results are a consequence of the efficiency and robustness of the DGT approach that allows to quickly identify new available nodes close to peer’s geographic location, by means of periodic discovery and maintenance procedures, and at the same time to detect disconnected nodes.

A second simulative analysis aimed at measuring the robustness of the DGT overlay with respect to increasing values of vertical handover WiFi latency. We considered a rather pessimistic range ($L \in [1s; 8.5s]$) of latency values (if compared to the ones used in other papers [24]), in order to heavily test our approach, also taking into account that the latency is constant for all the duration of the simulation and for all peers. In terms of network coverage the simulated scenario is again Scenario 1, where all types of connectivity are available at the same time in the area of interest. As expected, an increased value for the WiFi vertical handover latency heavily affects RT values (as shown in Fig. 5.42c), and consequently the global percentage of missing nodes (Fig. 5.42a), that proportionally grows with the latency. The same behavior can be observed for the number of lost packets per minute (Fig. 5.42d) that however remains globally small. As presented in the analysis related

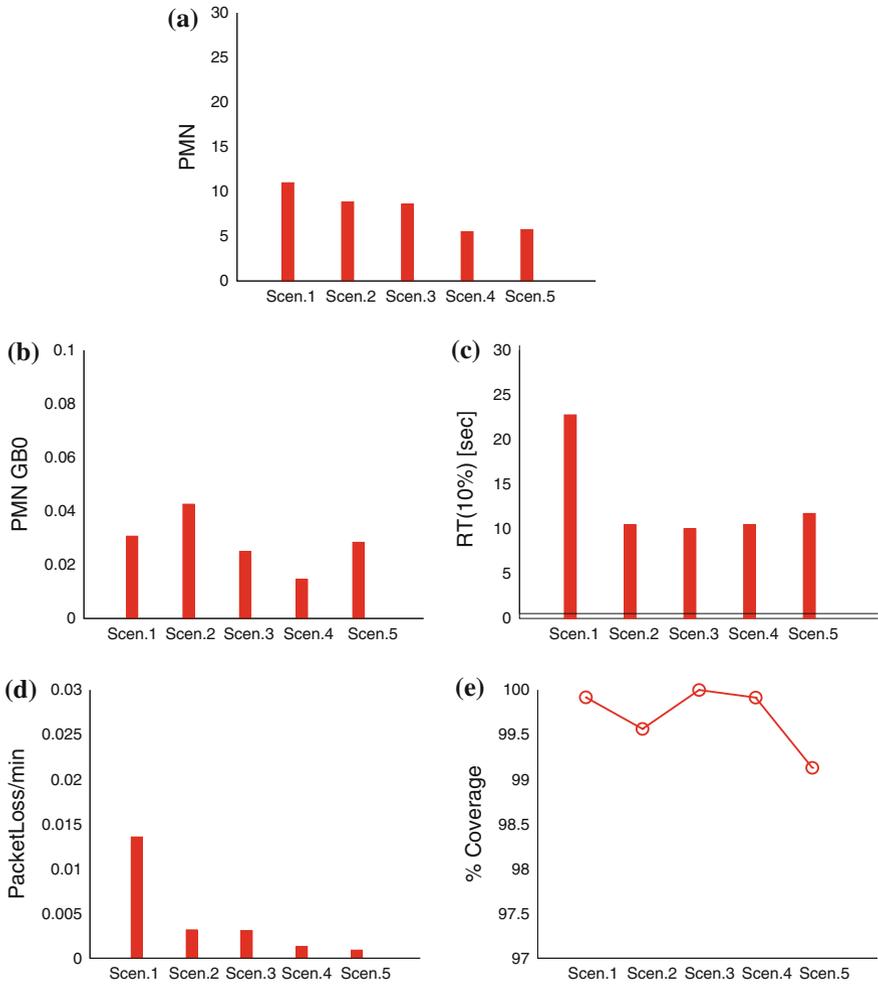


Fig. 5.41 Results related to different simulated scenarios. **a** PMN. **b** PMN in GB0. **c** RT(10 %). **d** PacketLoss/min. **e** % coverage of traffic information messages

to the variation of the connectivity coverage, one of the main important metrics for evaluating the robustness of the DGT approach is the PMN evaluated in the first GeoBucket(s). In fact, a high knowledge in the inner container and a gradually reduced value in the others mean that in any case the peer can perform successfully the discovery procedure, keeping the neighborhood updated. Most importantly, the peer can properly disseminate traffic information messages to its neighbors. Results presented in Fig. 5.42b, c confirm how the design of this peer-to-peer inter-vehicular DGT network is robust also in presence of a high values of latency, being able to correctly deliver messages with a percentage always higher than 99 %.

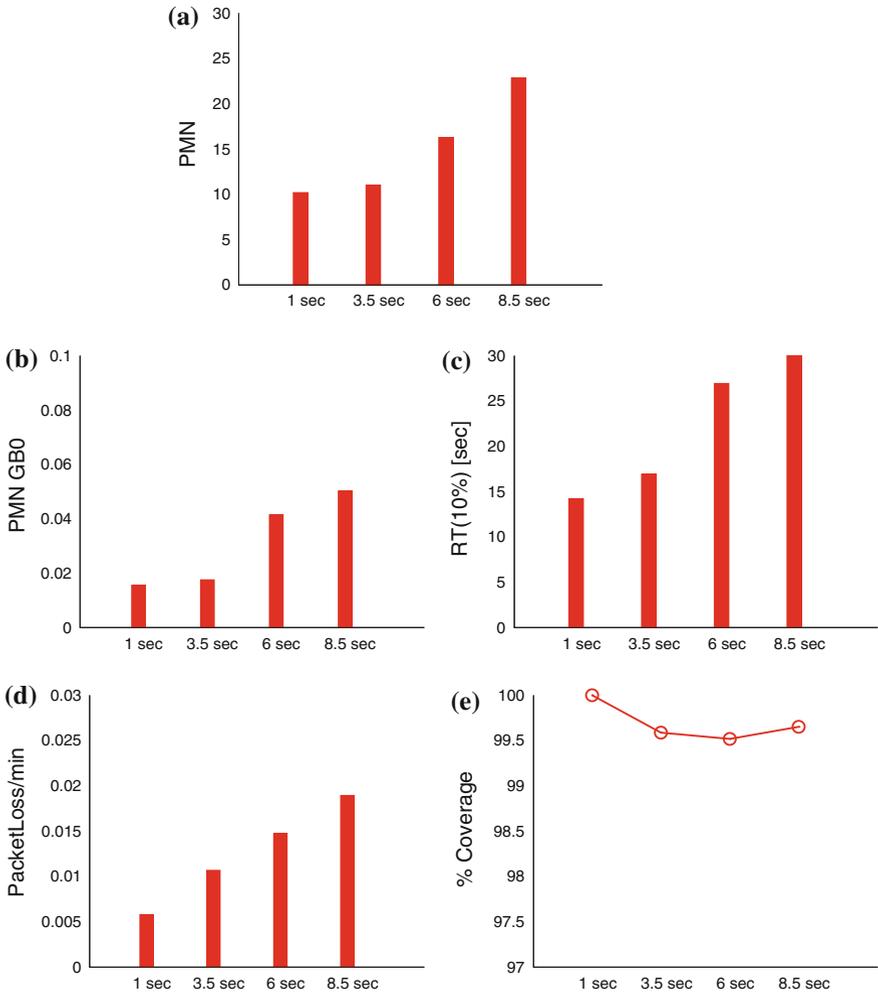


Fig. 5.42 Results related to different latency values. **a** PMN. **b** PMN in GB0. **c** RT(10 %). **d** PacketLoss/min. **e** % coverage of traffic information messages

5.6.2.7 Discussion

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

- Target Area A (as covered by the geo-buckets),
- Discovery Period (T_d),
- Position Update Threshold (ϵ).

How to configure optimally these parameters ultimately depends on the application. In this first analysis 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 frequent and often 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 hundred meters does not make any practical difference.

We also presented an investigation on the robustness of our DGT-based localization protocol to vertical handover in highly serviced urban areas. We have illustrated some significant simulative scenarios whose results evidence the independence of the percentage of missing nodes in the inner GeoBucket from peer disconnections due to vertical handovers as well as the short time required subsequently to recover knowledge about most neighbors consequently allowing to correctly deliver

Presented results are relevant also to show how a traffic information system and/or a vehicular network based on mobile devices and a peer-to-peer approach is feasible, and how in the near future those kinds of systems could be really and massively utilized by end users.

5.7 DGT for Vehicular Networks: The D4V Architecture

In this section we present the application of the DGT approach in the context of a completely distributed intelligent transportation system. We used the designed overlay to build a distributed traffic information system (TIS) for the dissemination of traffic

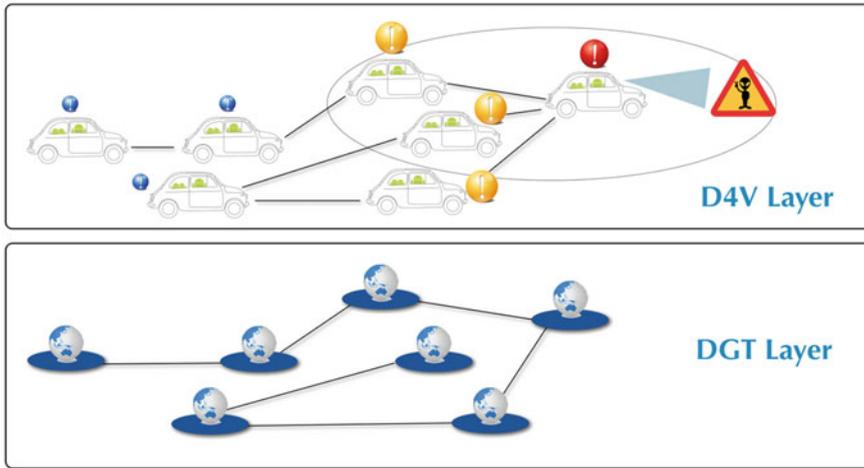


Fig. 5.43 DGT and D4V layers used to realize a distributed TIS

alert messages. In such a system (depicted in Fig. 5.43) users can participate using their smartphone to send and receive real-time information about traffic conditions or potentially dangerous situations. Furthermore, we present the implementation of the first DGT prototype to evaluate the performance of the protocol in a actual smartphone-based vehicular network.

5.7.1 Traffic Information System and Vehicular Sensor Networks

Driving safely, efficiently and comfortably does not depends only on the vehicle, but also on a large number of external factors that are difficult to predict without the support of IT. Among others, Vehicular Inter-networking [13] has a prominent role, paving the way to several valuable applications, namely geocasting, mobile data sensing and storage, street-level traffic flow estimation, etc. [33]. Vehicular networks builds upon Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) connection, as well as on hybrid variants [13].

Vehicular Sensor Networks (VSNs) are emerging as an appealing technology for monitoring the physical world, especially in urban areas where a high concentration of vehicles equipped with onboard sensors is expected in the near future. One reason of the interest for VSNs is that vehicles can be easily equipped with powerful processing units, wireless transmitters, and sensing devices. The latter may even be somehow complex, costly, and heavy like GPS, cameras, vibration sensors, acoustic detectors, etc.

Recently, the VSN research community has started investigating the possibility of using smartphones as V2V and V2I communication nodes, but also as portable sens-

ing platforms [34]. Smartphones are characterized by ever increasing technology—in terms of computational, networking and storage capabilities—and good connectivity. Users often carry such powerful handheld devices in their cars to take advantage of multimedia playback, navigation assistance as well as Internet connectivity. In the near future, many vehicles may be exploited as mobile sensors to gather, process and transmit data harvested along the roads in urban and extra-urban environments, potentially encompassing multiple types of information ranging from traffic/road conditions to pollution data and others.

As a matter of fact, until support for ad-hoc WiFi connectivity or for the new Wi-Fi-Direct standard⁵ will be widespread, smartphone-based VSNs will require the presence of a communication infrastructure (e.g., 3G/LTE cellular networks, WiMax, etc.). Thus, they will share the advantages of V2I schemes over V2V technologies, namely a better support in Commercial Off-the-Shelf (COTS) equipment, native long-range communication capabilities as well as support for broadcast or multicast communication (at least at the application level). At the same time, cellular-network VSNs exhibit some disadvantages w.r.t. V2V schemes, such as higher latency at short distances, local communication obtained only indirectly and by adding overhead, and the need for service coverage along with the associated data traffic costs.

While hybrid communication schemes, i.e., combining V2V with V2I capabilities, would inherently provide the best overall solution (robust, efficient Internet-capable networking), we remark that the choice of an infrastructure-based communication does not constrain the upper level of data dissemination and processing, as well as the application level of the service organization of traffic information systems. In fact, a V2I infrastructure does not necessarily imply a centralized organization, which would inevitably lead to scalability issues, for example to cope with the information requirements of thousands or millions of vehicles moving around in a large metropolitan area. While multiple distributed organizations (e.g., hierarchical) can be deployed to achieve better scalability, a completely decentralized P2P approach is highly appealing. Initially exploited within V2V schemes [1], P2P approaches have been recently followed also for realizing decentralized traffic information system [2]. In fact, the P2P application layer supports the decentralization of the responsibility as well as of the computational and communication loads, which can also be beneficial for smartphone-based VSNs [3].

In this context, we introduce the D4V (DGT 4 VANET) architecture, whose objective is to define a scalable architecture for opportunistic dissemination of data provided by vehicle sensors and drivers, relying on commercial smartphones, rather than dedicated devices. D4V is based on the concept of Distributed Geographical Table (DGT), to properly manage and update information about the neighborhood of a vehicle involved in the system.

An almost complete overview of existing and emerging technologies and solutions for distributing and aggregating sensor data in vehicular networks has been proposed by Uichin and Gerla [33]. An example is MobEyes [35] that proposes a strategy for harvesting, aggregating, and distributing sensed data by means of proactive urban

⁵ http://www.wi-fi.org/Wi-Fi_Direct.php.

monitoring services provided by vehicles that continuously discover, maintain, and process information about events of the urban scenario. Messages and summaries are routed to vehicles in the proximity, to achieve common goals, such as providing police cars with the trajectories of specific “target” cars.

A fundamental issue for VSNs is connectivity: different wireless access and communication methods have been evaluated, including Dedicated Short-Range Communication (DSRC) [36], WiMax/802.16e [37], WLAN [38], as well as cellular systems [39]. The use of a cellular communication network reduces the problem of implementing a working traffic information system (TIS), but introduces, on the other side, the issue of collecting data and distributing them to interested users.

A common approach is based on the client/server paradigm, where all data generated by vehicles are stored in a central server or a server farm on the Internet. Hull et al. [40] pointed out the major technical challenges of this solution, that are mostly related with the huge amount of simultaneous updates and queries generated by movements and requests of users (each car is a source of queries and sends its own measurements regularly).

For these reasons, in recent years researchers started investigating architectures based on the P2P paradigm, to build a distributed TIS where cars are not only consumers but also producers of information. Rybicki et al., with Peers on Wheels [3] and, more recently, with PeerTIS [41], have shown P2P architectures where participating cars are peers organized in a Distributed Hash Table (DHT). Roads are divided into road segments, each with a unique ID that is used as key in the DHT. The main idea is that each node is responsible for a certain part of the ID space and, consequently, for a certain number of road segments. Up to now one of the troubling issues is the fact that obtaining full information about planned and alternative routes is expensive in terms of bandwidth consumption. The work of Santa et al. [2] shows another P2P approach based on cellular networks (CNs) and on the JXTA middleware [42], to enable the transmission of information among vehicles and between vehicles and infrastructure, bounding the propagation of messages. CNs are also used not only in P2P solutions but also in participatory platforms, for participatory vehicular sensing [43–45], allowing applications such as ride quality monitoring, street-level traffic flow estimation, and proactive urban surveillance.

5.7.2 D4V

Most vehicular network safety applications need information from a very limited geographic area around the vehicle’s current position. This may not be the case for driving comfort applications such as traffic intensity or traffic jam monitoring, as well as parking discovery [46] or guidance systems that distribute information about the traffic or road state for the entire city or for those regions where the car is located or is moving towards. Our goal is to design and build a reliable and scalable system capable of disseminating in an opportunistic way information coming from

driver's inputs or directly from one of the vehicle sensors, e.g., active shock absorber, cameras, engine, temperature sensors, etc.

Generally speaking, distributing information over long ranges in vehicular applications is a very challenging task in terms of how to gather, transport and aggregate such information. In this Section, we consider the case of network and user interfaced uniquely by a mobile device. The information that enriches the knowledge base of the car will be collected from internal and external data sources, namely vehicle or roadside infrastructure sensors. The on-board intelligence of the car extends, maintains, and disseminates this information by creating a local view of the car surroundings.

In the literature different techniques for content dissemination in VSN are described, like flooding and geocasting [47, 48], request/reply [49, 50], broadcasting, *sharing* [51] and beaconing [52, 53]. In the D4V (DGT for Vehicular Networks) architecture, we adopt the DGT scheme and the opportunistic and spatio-temporal dissemination approach proposed by Leontiadis and Mascolo [54], which is based on the publish/subscribe paradigm and allows message distribution to all interested receivers in an area by keeping messages alive in that zone for a specified period of time. On account of its properties, we believe that such an integrated solution copes well with a very dynamic scenario where users can easily and frequently change their subscription interests according to their planned path, current season, city neighborhood, etc.

D4V's basic message has been defined with: a *type*, for the notification category (for example, the class of traffic events or sensor data); a *location*, associated with the information; a *range*, that represents the area around message's location that the notification should reach; a *time to live* of the event; and a message *payload* containing—whenever necessary—additional and detailed information about the event.

Different types of messages can thus be distributed by the same dissemination protocol. It is possible to create, for example, a message to warn approaching users about a traffic queue or a dangerous situation, to distribute data extracted from the different sensors of the vehicle or to notify other users about a free space in a parking area. Each user selects the list of message types for which she/he is interested and adds this information to her/his peer descriptor, allowing other peers to send only appropriate messages according to the receiver's preferences.

When a new message is generated, the publisher picks up from its GBs the closest known nodes within the notification's range that are interested in the particular information type (by reading the peer descriptor), and sends them the new message, trying to avoid duplications. When a notification is received, the system checks if it still matches the user interests or if it does not (in the presence of dynamic subscription), or if it is already known. In the case of a new information, the node adds it to its knowledge and distributes it again to known interested peers.

When a peer receives the references about a new node in its area of interest, it checks if in its knowledge base there are notifications not yet expired that may be useful for that peer. If the target peer has not yet been contacted for the same reason, the node sends the message. During this dissemination processes, it is necessary to check if some messages have expired, and, consequently, to remove them and their

references from the vehicle knowledge base, thus avoiding the distribution of an obsolete notification.

5.8 D4V Simulation

In this section, we present the simulative analysis of a D4V-based application that allows vehicles to adapt their routes according to traffic information gathered from other vehicles in the area. The following set of performance metrics are considered:

- *Bandwidth* [(dimension: (kbyte/peer · s)]: Average message rate sent per peer per second.
- *CP*: Estimated coverage percentage of D4V messages (*TrafficInformation* and *SensorData*) at a certain time of the simulation. It is evaluated as the ratio between the number of peers that actually received a specific message and the number of those which should have it.
- *TJCP*: Average percentage of cars involved in a traffic jam.
- *DFE* [dimension (km)]: the Distance From Event is the average distance between the geographic location of a vehicle that did not receive a traffic jam message and the position of that event. This metric improves the information provided by CP. Indeed, a high DFE value (compared with the message range) means that drivers who do not receive the message are far from the dangerous situation and probably will receive the information shortly from the other neighbors that have been already informed.

By means of DEUS [30], with the packet delay model presented in Sect. 5.6.1, we have simulated a VSN deployed across the city of Parma, considering a number of vehicles that move over 100 km of realistic paths generated using the Google Maps API. Each simulated vehicle selects a different path and starts moving over it. Using the features provided by the Google API we have created a simple HTML and Javascript control page that allows the monitoring of the temporal progression of the simulated system, in which any node can be selected to view its neighborhood (videos are available at <http://dsg.ce.unipr.it/d4v>). The simulated D4V covers ten hours of system life (10,000 virtual time units) with 20 switch stations, 5 virtual tracks with bad road surface (either ice, water, snow, or pothole), accident events scheduled during the simulation according to a Poisson stochastic process and with different message types to disseminate information about sensed data and traffic situation. Simulations have been repeated with 5 different seeds for the random number generator, that are sufficient to obtain a narrow I_{95} confidence interval ($\pm 10\%$ of the steady state value, in the worst case).

This simulation takes into account the configuration of GBs that was shown to obtain the best DGT performance in [4]. Each node has 4 GBs with a thickness of 0.5 km and a peer limit of 10 nodes, covering a region of interest of 12.5 km² and a dynamic discovery period ranging from 1.5 to 6 min depending on the number of discovered nodes.

The first step of the DEUS-based evaluation aims at analyzing the effect of varying the ε threshold ($\varepsilon \in [0.1; 1.0]$ km with a step equal to 0.3 km) considering two different vehicle densities $\delta = 10$ vehicles/km and $\delta = 20$ vehicles/km and a range of interest, for the disseminated messages, equal to 4 km. As defined earlier, ε represents the minimum displacement threshold considered by a peer to notify its geographic position update to nodes in its neighborhood. The analysis aimed at evaluating the effects of the variation of the update frequency on system performance and on information dissemination. Graphs in Fig. 5.44 show simulation results for the considered main metrics, in particular Fig. 5.44a illustrates the global CP as a function of ε values showing that traffic information messages are highly distributed to active peers in the configured range of interest. A higher peer density contributes to increase knowledge sharing, supporting the dissemination process given that more nodes receive and forward messages to interested drivers. In Fig. 5.44b the percentage of vehicles involved in a traffic jam is shown as a function of ε . The results confirms the robustness of the system which, even in the presence of a reduced update frequency, is able to properly distribute traffic information, leaving unchanged the percentage of drivers involved in the jam. The effectiveness of the approach with different position update thresholds can be observed also by analyzing the DFE value (Fig. 5.44c) that remains constant and very close to the dissemination range value of 4 km, confirming that peers that do not receive a traffic message are those located very far from the traffic event, thus having a good margin to receive the alert on time. This analysis suggests that vehicles involved in the queue are those that were really close to the traffic jam and had not enough time to react and change direction. Data traffic as a function of ε values is illustrated in Fig. 5.44d which confirms that finer position updates to neighbors yield to an increased network usage.

The second step of the evaluation has been driven by the goal of studying the D4V performance with respect to the variation of the dissemination range which is representative of the circular area around the message origin the notification should reach. Thus, in this scenario we vary it from 1 to 10 km to understand how it influences the dissemination process and its cost. Figure 5.45a, b illustrate the coverage percentage and the total number of vehicles involved in traffic jams, respectively. Both graphs show how a range of interest as small as 1 km affects the message distribution process due to a lower margin between the traffic jam and the drivers. In this situation, peers may receive alert messages when they are too close to dangerous situations, thus becoming involved in the queue. In particular, we remark how a lower vehicle density worsens such phenomenon because of the smaller number of nodes which can redistribute their knowledge about the traffic conditions. At the same time, it can be observed how there is no significant gap using range values larger than 4 km for both peer density curves. In Fig. 5.45c, the DFE value for the considered configurations is shown as a function of the range. For comparison, the optimal distance from the event is also shown. The latter coincides with the value of the dissemination range, because, ideally, the minimum distance of peers which did not receive the traffic information message yet is clearly the range of interest.

Results show that within a 4 km range the DFE remains close to the optimal bound, while higher values of the dissemination range decrease the DFE albeit still quite

Fig. 5.44 Simulation results for different ε values:
 Coverage % **(a)**. Number of vehicles in traffic jam **(b)**.
 Distance from event (km) **(c)**.
 Bandwidth (kbyte/peer/s) **(d)**

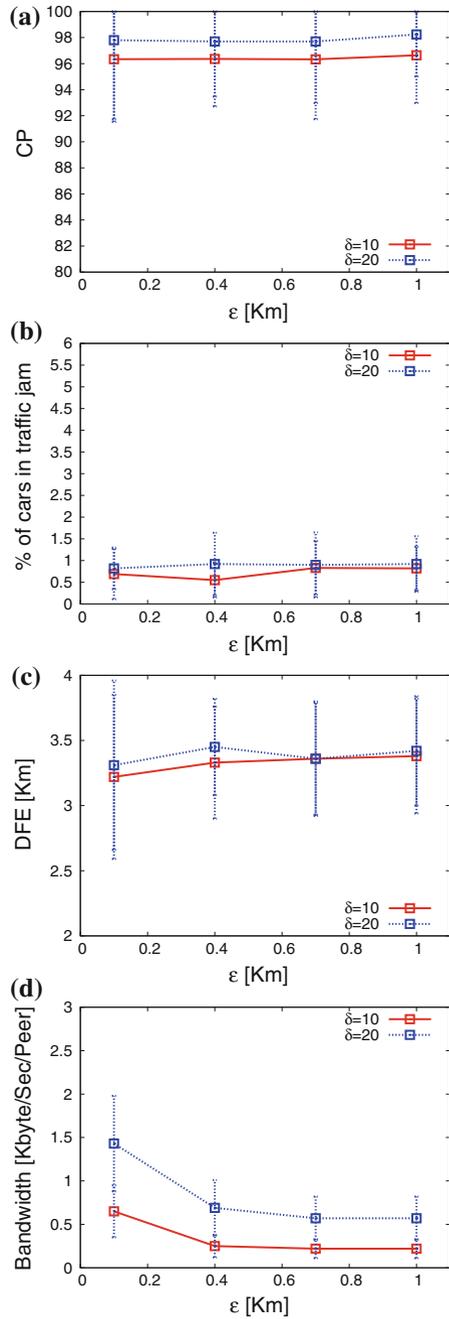
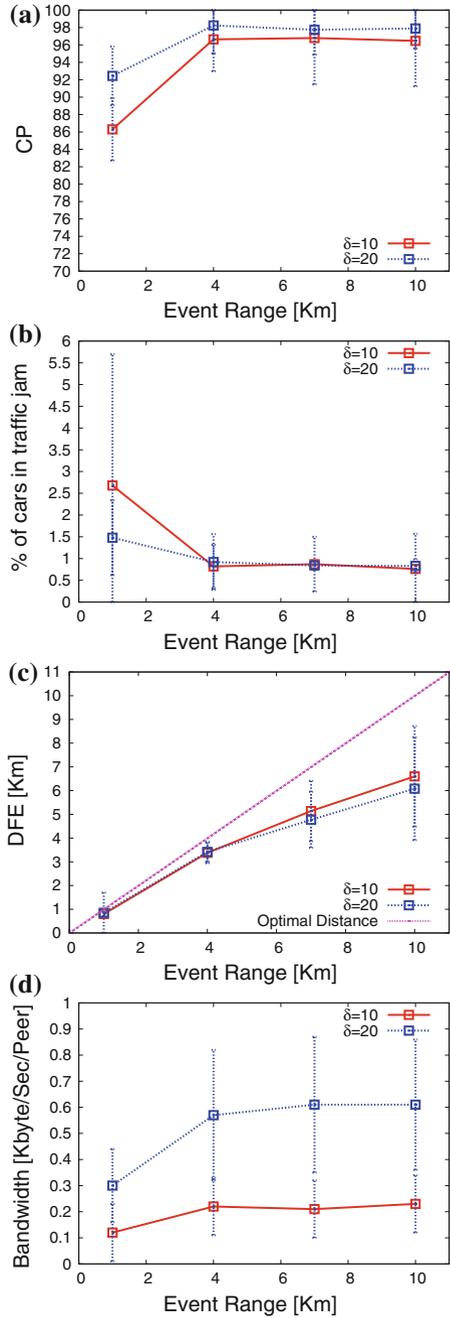


Fig. 5.45 Results for different values of the dissemination range: Coverage % **(a)**. Number of vehicles in traffic jam **(b)**. Distance from event (km) **(c)**. Bandwidth (kbyte/peer/s) **(d)**



close to the bound. Hence, drivers who do not receive the alert for a specific event are in any case sufficiently far from it and will receive the alert with enough time to react. Finally, an extended event range corresponds to an increased notification area and, consequently, to a larger number of interested drivers that may be contacted. However—as shown in Fig. 5.45d—this slightly affects the amount of exchanged messages.

The third stage of the simulative analysis aims at evaluating the system performance with respect to the variation of the peer density in the vehicular network. The dissemination range of each node is set to 4 km (chosen according to previous results) and the same dynamic discovery period presented earlier. The scenario is characterized by an initially growing number of active vehicles, followed by a stable phase without new joins or disconnections. The results in Fig. 5.46a confirm that the proposed solution is able to cope with different node densities with no performance degradation, keeping the Coverage Percentage value significantly high (between 98 and 100 %)—even in the case of very low density (5 peers/km) which could be quite critical for VANET-based applications. We recall that, if a mobile peer finds itself in a desert area, it will still be able to fill its external geobucket with remote peers, by requesting their contacts to the bootstrap node. This distributed knowledge provides appropriate support to efficiently disseminate messages about traffic jams or sensed data. As in the second experiment, the results in Fig. 5.46c show that an increasing number of active peers maintains the DFE high and close to the dissemination range. This results into an accurate dissemination of traffic information messages that allows drivers to receive alert information on time, still sufficiently far from the dangerous location.

In Fig. 5.46b, the percentages of vehicles blocked in a traffic jam, with and without D4V content dissemination, are directly compared. This confirms that the D4V approach drastically reduces the number of involved vehicles that would otherwise grow significantly for increasing density.

Figure 5.46d shows the average data traffic per peer (in KB/sec/peer) which is necessary to maintain the DGT overlay and disseminate traffic information messages to other active neighbors, as a function of the density. Since we assume to use UDP as transmission protocol, there are no retransmissions in case of lost packets. Here we show the average bandwidth—estimated from simulative results, corrected considering the cost of the headers—consumed in the best case (when the transmitted message is much higher than the IP header) and in the worst case (when the transmitted message has a size that is comparable to the IP header, e.g., a location update, that contains only a peer descriptor and a location). Even if there is a moderate and natural increase associated with the growth of nodes, the amount of data exchanged by each peer remains very limited. This behavior is associated with the fact that, as described in the previous section, D4V is based on an opportunistic content dissemination strategy. D4V tries to minimize the amount of sent packets, by forwarding them only to interested users, trying, at the same time, to reduce the number of duplicated messages. The density values we considered are 5/10/20 vehicles per km. Higher values would be neither realistic nor interesting, as they would mean that all vehicles on the roads are running the DGT.

Fig. 5.46 Simulation results for different peer densities: Coverage % (a). Number of vehicles in traffic jam (b). Distance from event (km) (c). Bandwidth (kbyte/peer/s) (d).

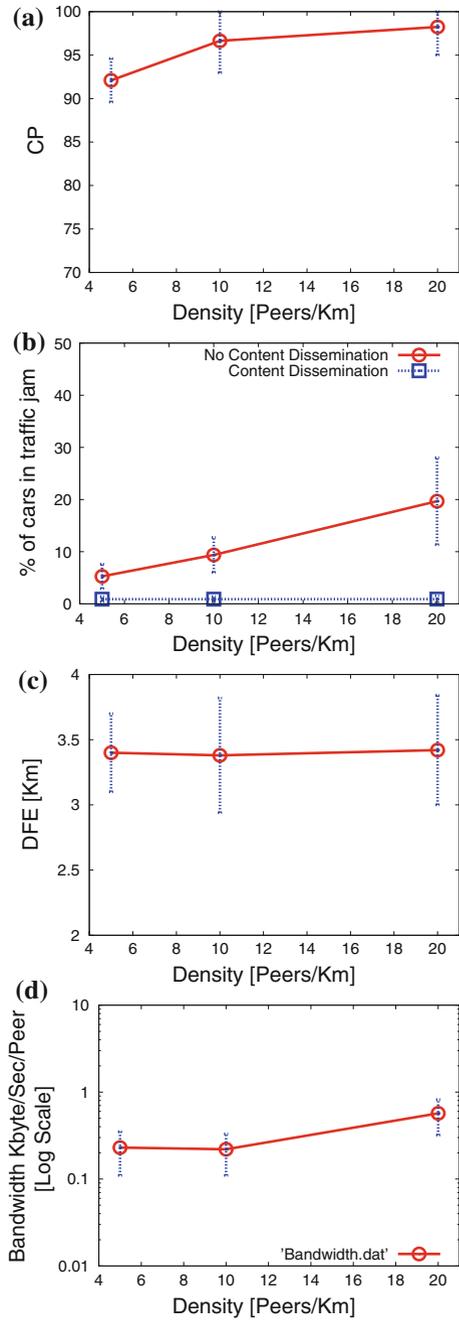


Fig. 5.47 Results for different Packet Loss Percentage: Coverage % (a). Number of vehicles in traffic jam (b). Distance from event (km) (c). Bandwidth (kB/peer/s) (d)

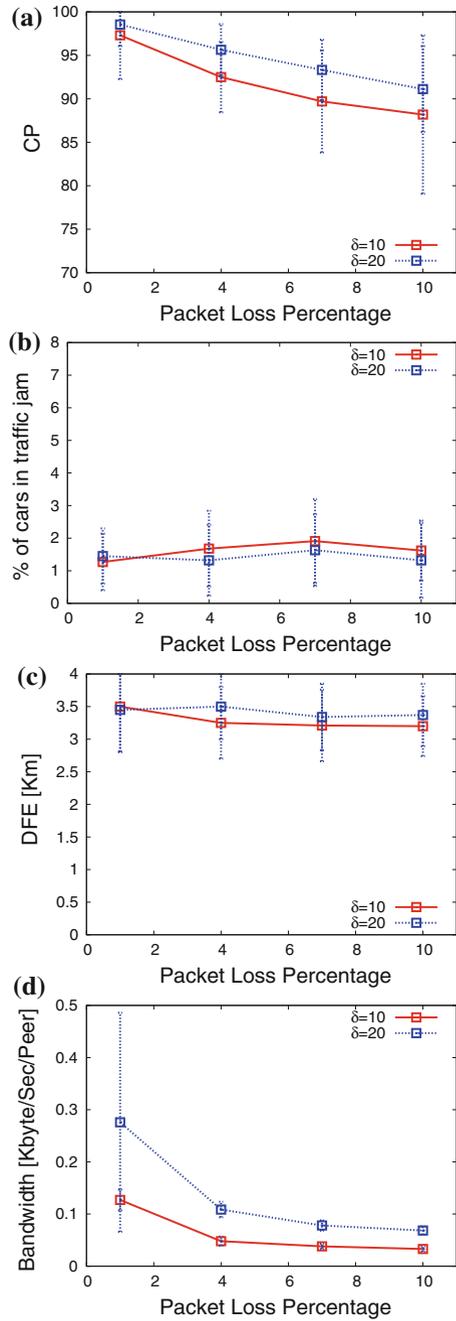


Figure 5.47 is dedicated to the analysis of the system robustness, in terms of packet loss percentage P , in a scenario with $\delta = 10$ peer/km and $\delta = 20$ peer/km, 4 GBs with thickness of 0.5 km, and a message dissemination range of 4 km. In the current form of the simulation code, there is no recovery procedure to verify whether a transmitted message is correctly delivered and, if it is necessary, to retransmit it. This is really important to properly interpret presented results, in particular for the dissemination of traffic information messages and the global robustness of the DGT approach. In Fig. 5.47a the global CP is shown as a function of the packet loss percentage, confirming that on average peers maintain a detailed knowledge of traffic events (>90 %) in the first GB.

In Fig. 5.47b, the percentage of vehicles involved in a traffic jam is investigated; it is a slightly increasing function of P , given that some peers may not receive alerts on the dangerous event and could be involved in the queue. The design and the distributed knowledge provided and maintained by the DGT allows to inform a large number of drivers keeping the number of queued vehicle really low. D4V robustness is also confirmed by the results in Fig. 5.47c, showing that the nodes that do not receive traffic information messages are considerably distant from the event location. Moreover, the DFE is almost independent of P . Figure 5.47d reports the data consumption, which is unavoidably lower than in the other scenarios due to the lack of a recovery procedure for lost packets.

Finally, considering the behavioral model of a driver in proximity of a road stretch with a bad surface condition, that we have introduced in Sect. 5.5, in Fig. 5.48 we show the monitored speed for five virtual tracks with bad surface conditions for all drivers (including both the informed ones and those not informed) that drive across the street during the simulation. The observed results clearly show that a decreased speed is measured near the critical location (at distance zero), along with an increasing velocity while moving away from it. Because of that, we can say that the deployment of D4V would probably reduce the risk of accidents and nuisances along troubled roads on account of the achieved information sharing among drivers, including those still approaching the dangerous point.

5.9 D4V Prototype

The simulative analysis of sample scenarios based on experimental measurements of coverage and connection throughput, carried out across/around Parma urban area, gave us valuable insights to start the development of the first release of a DGT Library and the first prototype of the D4V system. The library implements the base functionalities and policies of a DGT overlay such as the discovery procedure, the management of the neighborhood and the GBs maintenance. The D4V application layer uses such features to implement the content dissemination algorithm and the user interface to get the input from the drivers related to a specific traffic event, and to show approaching dangerous situations. The development of the DGT library started from our novel peer-to-peer middleware called *Sip2Peer* [55], which is an open-source SIP-based

middleware for the implementation of any peer-to-peer application or overlay without constrains on peer nature (traditional PC or mobile nodes) and specific architecture. At this moment sip2peer is available for Java SE and Android platforms, but we are working on the iOS release. The Java and Android implementations of the library are based on the Java SIP stack called *MjSip* [56] that allows to manage the exchange of SIP messages to control multimedia streams. Sip2Peer supports two message formats. It is possible to manage simple text message containing any kind of information like raw data or XML, and it is also possible to natively use the JSON format. Following this scalable approach, the main class of the Sip2Peer API, i.e., Peer, provides all necessary methods for sending and receiving messages, allowing the developer to select the best solution according to his/her protocol and overlay and solve problems related to NAT traversal.

Figure 5.49 groups together DGT and D4V modules, to present all involved elements whose integration defines the behavior of a peer. Such modules are presented and described in the following.

- *Sip2Peer Layer (SL)*: Represents the communication module providing methods to receive and send messages from and to other active peers in the network. This layer interacts with the DMH (illustrated below) to route and forward outgoing and incoming messages, providing proper notifications when a packet has been correctly delivered or not.
- *DGT Message Handler/Dispatcher (DMH)*: Conveys and manages all DGT messages. It does notify neighbors position updates and redirect subscriptions and event messages from and to the Subscription Manager (detailed below).
- *Geo-Bucket Manager (GM)*: Manages the data structures of the peer according to its geographic location. This module interacts with the Location Manager to be notified for a position update, and thanks to DMH notifications it is able to add new discovered peers or remove nodes out of the region of interest. Since it is a

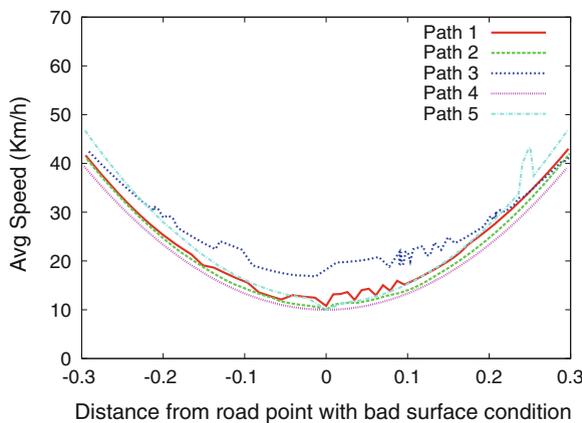


Fig. 5.48 Average of driver speed near road points with bad surface condition

base module of the DGT Library, it is configurable allowing to define the number of buckets and their thickness (target area) and the maximum number of nodes that a single GB can maintain during node life.

- *Location Manager (LM)*: Subscribes to information and updates about device location through GPS, WiFi or cellular network, trying to minimize the energy consumption according to the context application characteristics. In detail in a urban scenario where the mobile is used inside a car where the energy consumption is not a constrain the location could be detected used external localization devices such as a navigator providing that potentially could provide additional and useful information about the planned route and the target location.
- *DGT Kernel (DK)*: Is the core of a DGT node, implementing the routing strategy and the discovery procedure for neighborhood maintenance, as well as short/long range queries triggered by the user (through the User Interface) or periodically efficiently scheduled by the DK thread according to the application purpose and users settings. It also allows the interaction with DMH and GM to properly disseminate messages and alerts coming from UI or other external inputs.
- *Subscription Manager (SM)*: It is related to the D4V prototype and has been designed to manage the subscription system of the node, allowing to add or remove subscriptions and handling an filtering incoming events or user queries. It does interact with the UI to notify relevant incoming alerts or messages, and sets preferences about the subscriptions.

The User Interface (UI) allows to present to the user all required information and interface elements, to control DGT functionalities, like dissemination an alert messages about traffic jams or to schedule a query concerning a region of interest. It allows to visualize on a map (or in a dedicated list view) peer/vehicle and neighbor locations, as well as scheduled query results. Our first prototype and the associated UI has been designed and developed on the Android platform.

When the he/she runs the application, the user watch a map displaying the updated vehicle location, and configure through a specific menu the ip of the bootstrapping node used to join the DGT network (Fig. 5.50a). When the first DGT discovery has been completed and the neighborhood is formed the user can see neighbor vehicles on the map view and alert messages near its geographic position (Fig. 5.50b). For this first prototype we gave the possibility to the user to generate information related to four different type of event through a dedicated and simple user interface (Fig. 5.50c). By clicking on the associated button the user can generate and distribute an alert message related to Traffic Jam, Car Accident, Man at Work and Bad Surface Condition. The generated message contains information about the geographic location, the event generation time, the source user and the expiration time (set by default to one hour), after which the DGT peer stop disseminating it, unless a new user refreshes the information. While the user is driving, changing its location, the application checks if one or more received traffic messages are close to the car in a range of 200 m, and notify with a dialog message the alert type and its location (Fig. 5.50d, e). Furthermore, the user can review in any moment the list of received

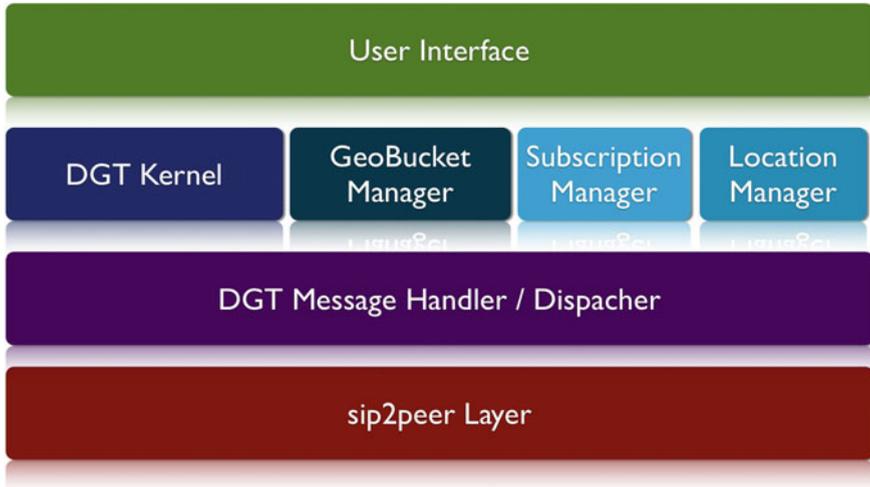


Fig. 5.49 D4V prototype modules

messages (Fig. 5.50f) sorted by distance, visualize them on the map, renew expired content or report an abuse or false information generated by a user.

Our plan was to evaluate the designed application during the development phase in a scenario with a relevant number of users before releasing the prototype to a group of alpha tester. To this purpose, an additional module has been created to emulate the behavior of vehicle moving along city streets. This module implements an FTM mobility model to evaluate the car speed, based on the switch station model presented in the previous chapter, and provides a web-based tool (Fig. 5.51) to monitor peer movements during experiments. Using this module has possible to create a complete and autonomous D4V node (called D4V-Bot) able to join the DGT network and generate a traffic message if required by the experiment setup.

5.9.1 Performance Evaluation of the D4V Prototype

A first evaluation phase has been conducted with a hybrid group of 50 nodes composed by real Android devices and D4V-Bots. The experiments have been conducted initially in a controlled environment of our laboratory (Fig. 5.52a, b). Peers were able to join the network, build their neighborhood and maintain it during the experiment, according to the changes of their geographic location. When a node (Android or Bot) generates a new message related to a traffic event, the latter was correctly distributed and shown on smartphones of user inside the region of interest of the event. Starting from the results of this first preliminary evaluation it was important to properly measure the network performance in order to understand if the results of the simulation

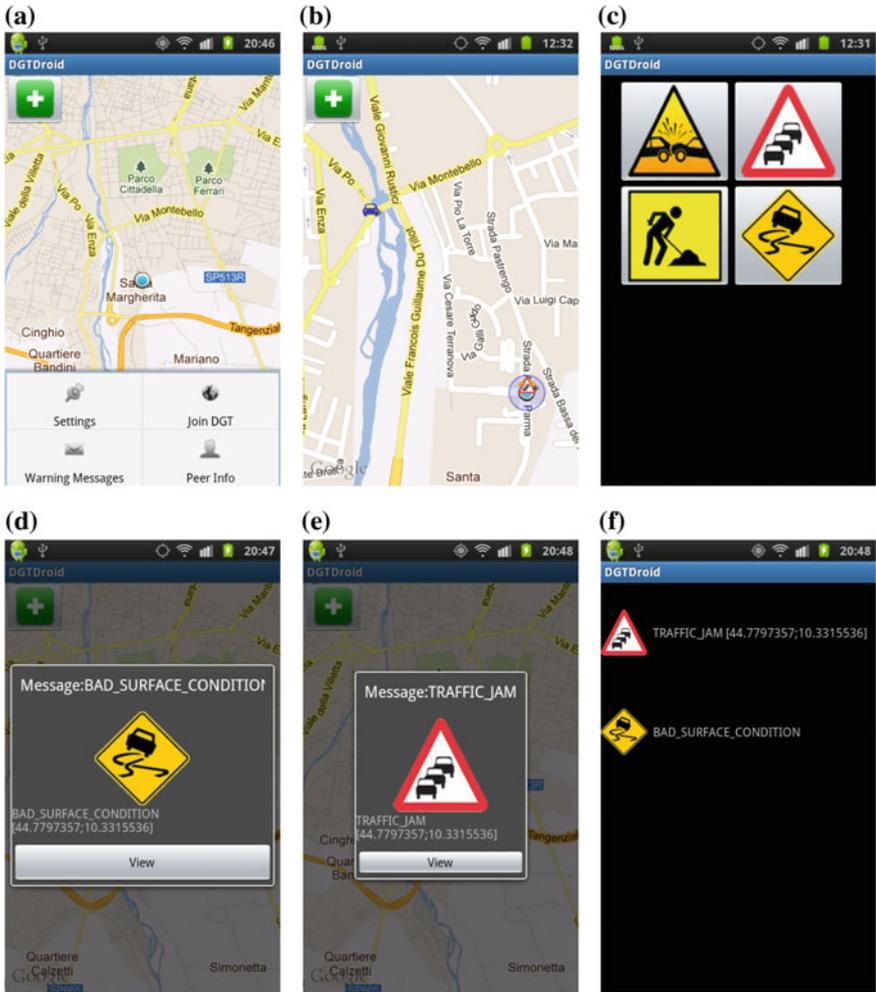


Fig. 5.50 Android DGT Prototype. a Settings menu. b Map view. c Traffic message creation view. d, e Bad surface condition and traffic jam messages popup view. f Incoming message list

analysis are confirmed by the results of the prototype test, also with an heterogeneity in terms of access network. For this purpose we deployed our D4V-Bot experiment on PlanetLab.

PlanetLab is a global research network that supports the development of new network services. Since the beginning of 2003, more than 1,000 researchers at top academic institutions and industrial research labs have used PlanetLab to develop new technologies for distributed storage, network mapping, peer-to-peer systems, DHTs, and query processing. PlanetLab currently consists of 1,089 nodes at 532 sites and

the University Of Parma contributes with 2 nodes in the world and european network. We deployed 50 D4V-Bots on 13 different PL servers in 13 different countries. Each node every 30 s logs on file a JSON string containing all the needed information to analyze the behavior the peer such as geographic location, exchanged kbytes, received and sent messages. At the end of the experiment a dedicated tool parses all available log files to build a time line of the experiment made by steps of 30 s containing all the required statistic for the performance evaluation.

Experiments results are based on five different runs of 26 min. The performance metrics that have been taken in account are:

- **Bandwidth** [dimension: (kbyte/peer · s)]: Average message rate sent per peer per second.
- **CP**: Estimated coverage percentage of D4V messages (*TrafficInformation* and *SensorData*) at a certain time of the simulation. It is evaluated as the ratio between the number of peers that actually received a specific message, and the number of those which should have it.
- **DFE** [dimension (km)]: the Distance From Event is the average distance between the geographic location of a vehicle that did not receive a traffic jam message, and

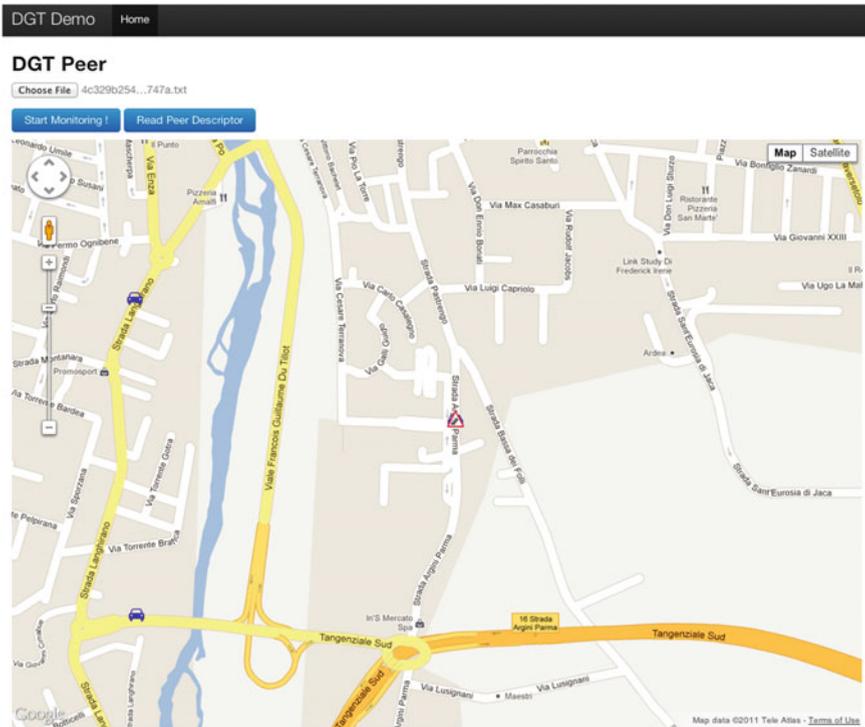


Fig. 5.51 D4V prototype web monitoring tool

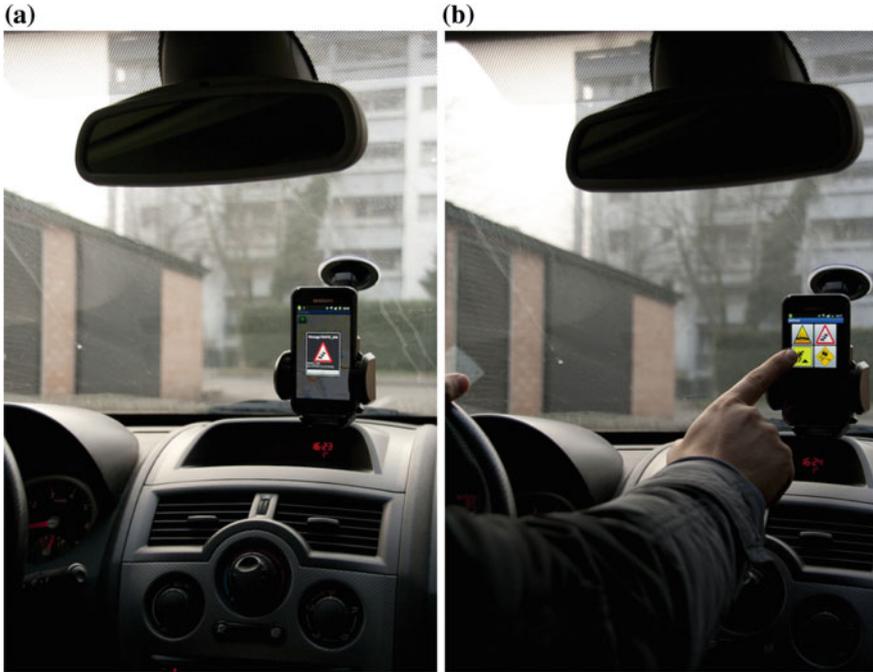


Fig. 5.52 Car setup. **a** Incoming warning. **b** Traffic message creation

the position of that event. This metric improves the information provided by CP. Indeed, a high DFE value (compared with the message range) means that drivers who do not receive the message are far from the dangerous situation, and probably will receive the information shortly from the neighbors that have been already informed.

- *Delay*: Represents the round-trip delay time (RTD). It is the length of time it takes for a signal to be sent, plus the length of time it takes for an acknowledgment of that signal to be received. This time delay therefore consists of the transmission time of a signal between the two points of a signal.
- *%Packet Loss*: Average % of Packet Loss for a peer during the experiment.

The graph in Fig. 5.53 shows the trend of the Coverage Percentage with the average value for the PlanetLab experiment and the same results obtained in the simulation analysis of our previous evaluation. The generation of traffic messages starts 400 seconds the D4V-Bots have started running, in order to give them enough time to build the DGT network. Results show that the average value of CP is really high close to 97 % and in particular significantly near the average value of our simulations ($\cong 98$ %). The CP curve shows that when new messages are generated the coverage percentage goes slightly down to lower value ($\cong 88$ %) but after one or two time line steps (30/60 s) recovers to an high coverage percentage confirming

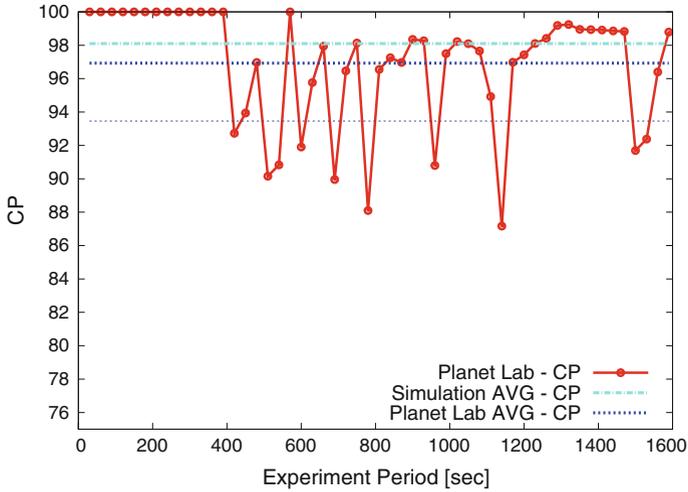


Fig. 5.53 PlanetLab—coverage percentage

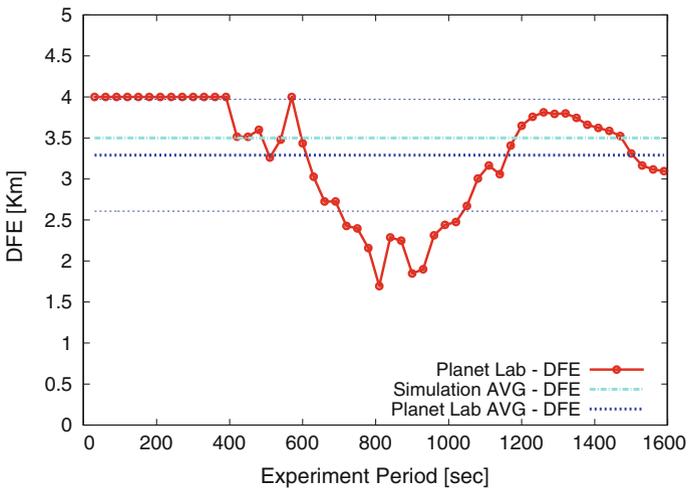


Fig. 5.54 PlanetLab—distance from event (DFE)

that the dissemination process and the neighborhood knowledge allow to efficiently distribute messages.

An additional performance metrics that allows to better understand the behavior of the protocol is the DFE values. For such an experiment we have considered a range of interest for disseminated message of 4 km. Figure 5.54 illustrates the DFE trend comparing PlanetLab and simulation average, that also in this case are really close and around ($\cong 3.5$ km). The graph confirms that vehicles that did not receive the message

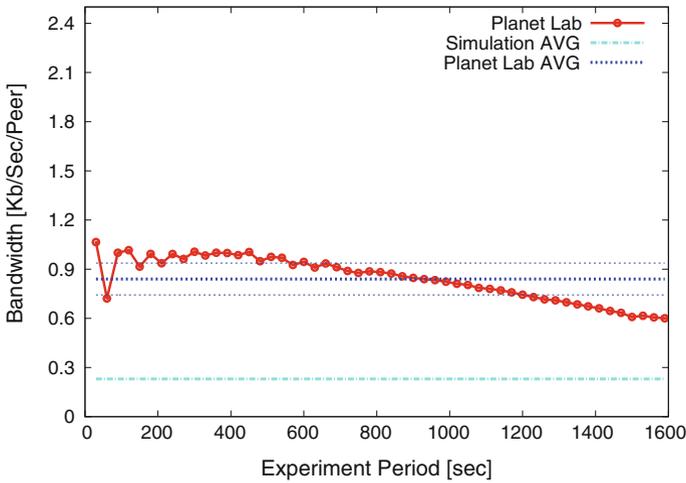


Fig. 5.55 PlanetLab—average bandwidth

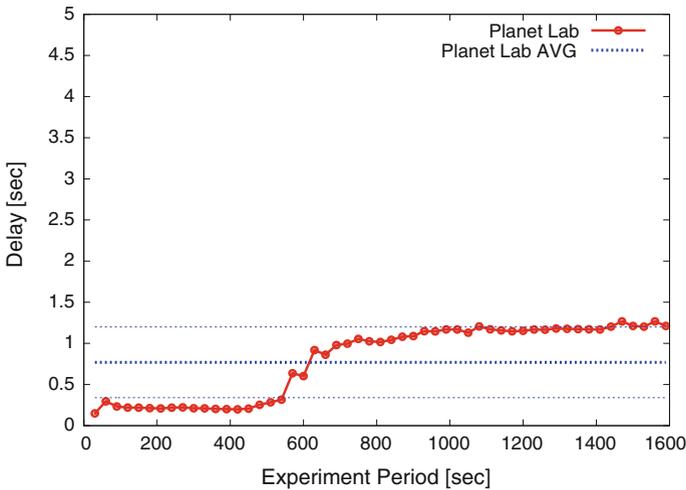


Fig. 5.56 PlanetLab—average delay

are on average really far from the dangerous event, and have a sufficient margin to receive the message before approaching the potentially dangerous location changing their direction to reach their destination using a different route or just adapting their vehicle speed for example near a portion of damaged road surface.

The graph in Fig. 5.55 reports the bandwidth in terms of kb/s/Peer during the experiments conducted on PlanetLab. In this case the average value ($\cong 0.3$ kb/s/Peer) is different from the same value obtained during the simulation ($\cong 0.9$ kb/s/Peer).

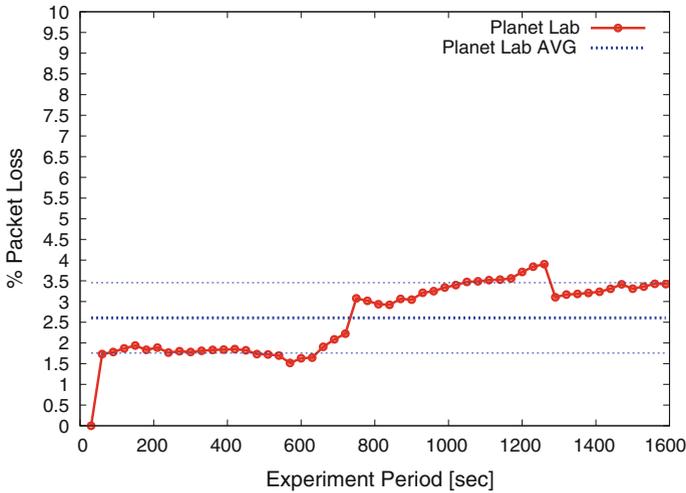


Fig. 5.57 PlanetLab—average packet loss

This difference can be related to the fact that in the real implementation there is an additional overhead due to packet header and additional exchanged information, that initially was not considered during the modeling of the communication in our simulator.

The graphs in Figs. 5.56 and 5.57 finally illustrate the trend and the average of delay and the percentage of packet loss measured during the experiments. The delay is reasonable for the designed application, considering that the small package size and that the experiments have been done on 13 servers located in different research institutions, with different network capabilities and load during the experiment (PlanetLab nodes are used at the same time by several application and the condition could change in the course of the experiment). After the evaluation on the PlanetLab network the next phase has been dedicated to implement and optimize the basic implemented library trying to reduce the amount of messages exchanged among connected nodes. The first optimization is related to the number of exchanged messages during the publication and discovery phases. Each `DGTEvent` message is attached with a list of size L containing the descriptors of peers which have already received that message. In this way, the message is forwarded only to nodes interested to it *and* not already receiving it. Moreover, the messages are forwarded by a peer to its neighborhood with a given probability p . It is straightforward to observe that the forwarding probability has to be properly set. In fact, if this parameter is too low, a given message may be stopped somewhere in the network, and no more propagated. Another implemented modification is a method to inform when a node is disconnected from the network. In the original solution, a node is recognized as disconnected if it does not reply to a presence message. In the modified solution, instead, if a node wants to disconnect from the network, it informs its neighborhood.

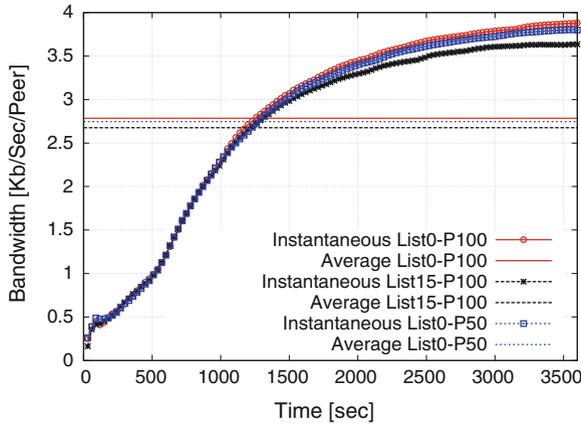


Fig. 5.58 Comparison of bandwidth consumptions for analyzed configurations

Finally, the DGT messages have been provided with a time to live (TTL), as well as the possibility to reduce the list size from the bootstrap node. Also these optimizations have been made to reduce the bandwidth usage. A final important feature, inherited from Sip2Peer, is the SBC-based NAT traversal mechanism, which allows the peers to communicate in presence of NATs—a very relevant problem especially in mobile scenarios. The SBC (Session Border Controller) is a node provided with a public IP address. It allows a generic peer to check if it is in a private network with a NAT, also requesting (if needed) a IP address/port pair to communicate with other peers.

The updated library has been tested both on traditional PCs and on real Android devices to validate the designed behaviors with different mobility patterns and network coverages. Consecutively, an additional evaluation of the DGT Java Library has been carried out by deploying 200 peers on a cluster at our Department. Each peer is randomly placed in the area of the city of Parma. All peers are connected to the DGT overlay network and can move according to the FTM mobility model described by (5.11). Performance evaluation is performed by letting each peer logging every 30 s its speed, geographical position, and amount and type of exchanged data. During network operations, another peer is generated and selected as the bootstrap node which generates random events, e.g., accidents or road works. Three scenarios have been analyzed, changing L and p . In the first case, $L = 0$ and $p = 1$; in the second scenario, $L = 15$ and $p = 1$; in the third scenario, $L = 0$ and $p = 0.5$. Finally, 5 independent runs have been performed to average over statistical fluctuations.

In Fig. 5.58, the bandwidth [dimension: (kB/s)] is shown, as a function of time, for the three considered scenarios. Both time-instantaneous and average values are shown.

One can observe that the bandwidth increases with time, since more and more packets and messages are generated during the experiments. For a fixed value of p ,

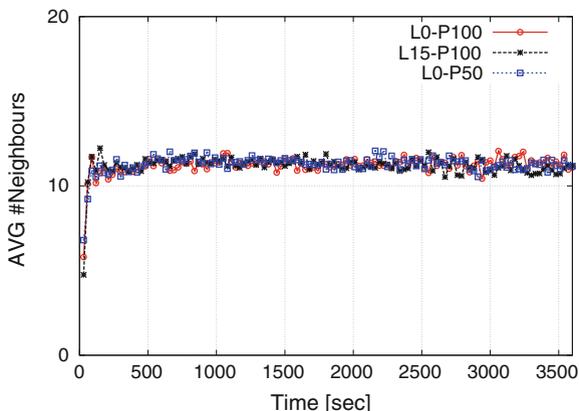


Fig. 5.59 The average number of neighbours for each active peer

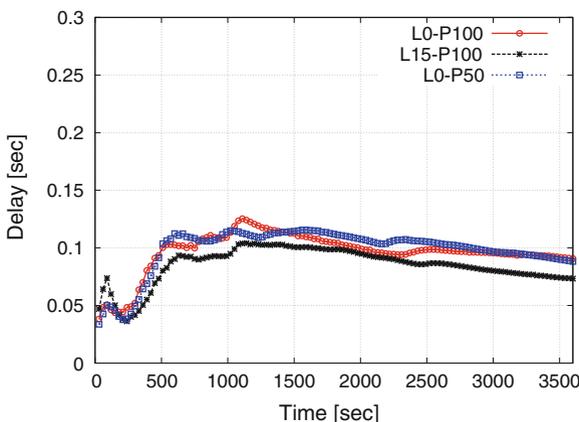


Fig. 5.60 The average delay measured during the experiment

increasing the value of L leads to better performance. In fact, each forwarding node is aware of more nodes which have already received a given message. Moreover, for a fixed value of L , decreasing the forwarding probability p reduces the bandwidth usage, due to the fact that less packets are transmitted across the network.

Since from Fig. 5.58 the lowest bandwidth usage can be obtained with $L = 0$ and $p = 0.5$, we now analyze the delay and neighboring performance in this scenario. In Figs. 5.59 and 5.60, the delay (D) and the number of neighbors (NN) are reported, as functions of time, for $L = 0$ and $p = 0.5$. Both time-instantaneous and average values are shown.

Note that both D and NN are stable around their mean value, thus showing that the effectiveness of the proposed DGT approach is confirmed in the experimental setting as well.

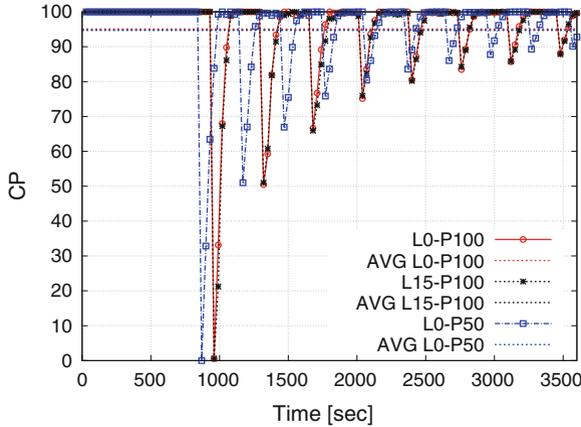


Fig. 5.61 Coverage percentage trends of evaluated scenarios

Graph in Fig. 5.61 shows the trend analysis of the Coverage Percentage during the experiment for the three considered scenarios in terms of p and L . At the beginning of the evaluation the measured metric is always at 100 % since there are not events in the system and the peers of the DGT network are just maintaining the geographic routing tables while they are moving according the implemented behavior and model. After 15 min a new node called Event Generator (EG), responsible for the generation of traffic events (selected among previously introduced types), joins the P2P network and when has built its initial neighborhood starts a periodic (with an average time $T_d = 4$ min) internal process to disseminate random event along the streets. Lower peaks in the graph trends are related to the instant of the generation of the event when the percentage of nodes that have received the alert in the area of interest is naturally low. At the beginning of the EG life the peaks are higher since the peer is still building and improving its geographic knowledge. Nevertheless the recovery time needed to bring the coverage percentage over the 90 % is significantly low and in the vicinity of a few number of time slots of 30 s (on average around 1.5 min to recover over the 90 %).

As previously illustrated, the coverage percentage is only the first aspect of the D4V analysis. In order to understand which is the distance from the event (DFE) of those peers that did not received yet the alerts generated by EG . The trends of the DFE metric in the evaluated scenarios illustrated in Fig. 5.62 confirms the excellent performance of the DGT overlay and its capability to properly disseminate geo-referenced messages according to D4V specifications. Reported values show how the average distance from the event during the overall experiment is significantly high (around 3.7 km) with only a lower peak of 1 km a the beginning of EG file (when its geographic knowledge is lower). Both presented graphs illustrate how the use of different values of L and p affect as expected only the bandwidth consumption and not the D4V behavior. CP and DFE have the same trend with negligible differences

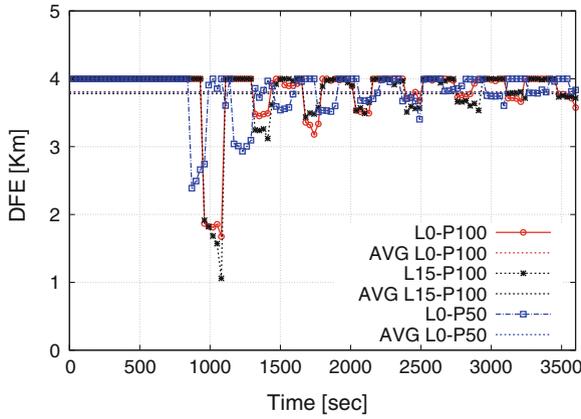


Fig. 5.62 Distance from event trends of evaluated scenarios

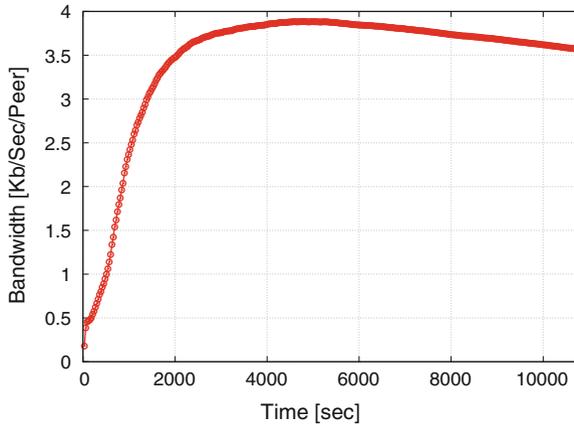


Fig. 5.63 Bandwidth consumption for an experiment of three hours with $L = 15$, $p = 100$ and 200 active nodes

attributable to the randomness of the mobility patterns and of the events generation that in a real evaluation are not easy to control.

At a later stage and in order to evaluate the developed library during a longer experiment, we have performed a new evaluation using the better configuration in terms of bandwidth consumption (with $L = 15$ and $p = 100$) and a duration of three hours with 200 nodes and a single event generator EG that joins and generates traffic related events after the first 15 min.

The first graph showed in Fig. 5.63 reports the average bandwidth during the experiment showing how it grows at the beginning while the DGT/D4V network is the initial phase and active nodes are improving and building their distributed geographic knowledge. Step by step the network reaches a stable state converging to

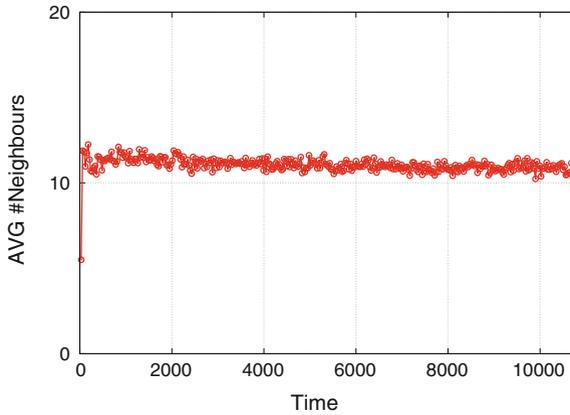
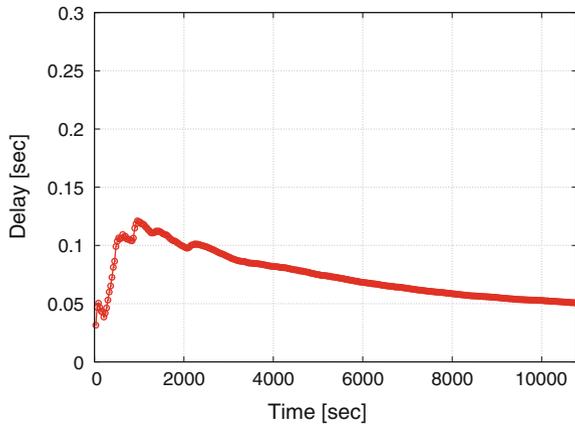


Fig. 5.64 The average number of neighbours for each active peer

Fig. 5.65 The average delay measured during the experiment of 3 h with $LA = 15$, $p = 100$ and 200 active nodes



a lower average bandwidth value. In any case showed bandwidth consumption are absolutely feasible for existing GPRS, UTMS and 3G cellular networks consuming only a small amount of the available capacity.

Graphs in Figs. 5.64 and 5.65, as in previous analysis, show respectively how the average delay and the number of neighbors are stable around their mean value also during a longer simulation confirming the effectiveness of the proposed DGT/D4V solution.

Figure 5.66 reports the trend of the average global coverage percentage during the performed evaluation and allows to confirm the behaviour identified with the previous analysis also during a significantly longer experiment. The graph shows how the *CP* has some lower peaks only at the beginning while the *EG* knowledge is still in an initial phase and step by step when the global distributed knowledge

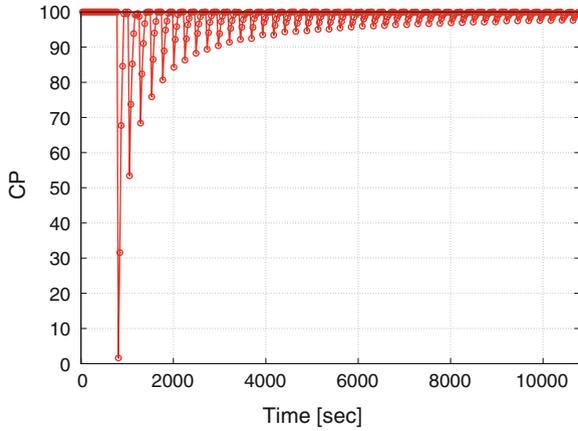


Fig. 5.66 Coverage percentage trends during the experiment of 3 h with $L = 15$, $p = 100$ and 200 active nodes

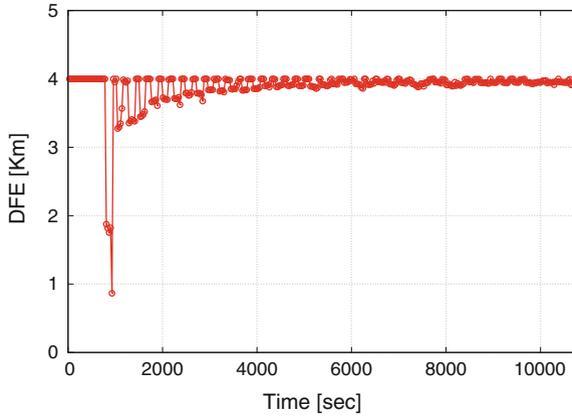


Fig. 5.67 Distance from event trends during an experiment of 3 h with $L = 15$, $p = 100$ and 200 active nodes

is higher the required time to recover the percentage over the 90 % is dramatically reduced and on average under 30 s.

This suitable and expected behavior is replicated in Fig. 5.67 showing the average distance from event of active peers in the network. It confirms the excellent performance to properly disseminate alert messages among interested and close neighbors and shows how the network is able to increase its effectiveness while the distributed knowledge is growing up. After an initial phase where the *DFE* metric has some lower peaks the convergence reaches remarkable values really close to the

limit of 4 km of the region of interest. Presented results completely depicts the performance and the behaviour of the designed and implemented overlay where active nodes can efficiently disseminate and receive traffic alert messages (through their smartphones) with reduced dissemination periods and with an high average distance from the potentially dangerous location giving them the time to change driving direction, reduce the vehicle speed and generally react and be focused on their driving.

5.10 Concluding Remarks

In this Chapter, we introduced and presented the design, implementation and evaluation of a novel structured P2P overlay scheme, called *Distributed Geographic Table* (DGT), which allows its (mobile) participants to efficiently retrieve node and resource information (data or services) located near any chosen geographic position. In a DGT network the responsibility for maintaining information about position of active users is properly distributed among nodes, for which a change in the set of participants causes only a minimal amount of disruption, without reducing the quality of provided services.

The following objectives guided the design of DGT:

- avoiding that a central node, or a particular peer, could become a bottleneck, or even a single point of failure, for the whole system because of its responsibility on a large subset of the network or of a specific geographic region.
- reducing the risk of disseminating obsolete information in the network;
- managing, by means of a distributed strategy, a high amount of simultaneous updates and queries, while limiting the platform cost—thus, making easier the access to the market;
- to build an overlay where neighbors in the P2P overlay are at the same time real geographic neighbors, thus allowing to avoid the need to forward additional messages to discover closest nodes;
- to properly evaluate and take into account the heterogeneity and mobility of user devices.

We evaluated the discovery procedure of DGT by means of analytical methods, and state-of-art simulation techniques, which enabled a high level of realism for the modeled scenarios. The DGT algorithms achieve convincing performance for extended ranges of system parameters, and properly maintain the neighborhood knowledge of each peer, discover new nodes and updated information with reduced overhead in terms of exchanged messages. All parameters in a DGT overlay can be tuned in order to achieve the most suitable trade-offs and performance, for the target application.

In the second part of the chapter, we illustrated a DGT-based TIS called D4V. In such a system, user nodes can be installed on smartphones, or in car equipment, to

send and receive real-time information about traffic conditions or potentially dangerous situations. The D4V architecture was evaluated by means of extensive simulations based on established vehicular mobility models, and through the deployment of a mixed network including both virtual and real users with Android devices.

A D4V prototype, based on the Sip2Peer open source middleware developed in our laboratory, has been deployed on the PlanetLab testbed, to verify network performance in a heterogeneous environment. Simulative and real measurements are very close, thus confirming the reliability of the simulation model, as well as the good performance of the DGT overlay. The scheme guarantees a high coverage, in terms of vehicular notification, over a wide range of system parameter values, whilst generating limited data traffic and well coping with significant packet losses. Hence, we are confident that D4V could be effectively used on the road to reduce the number of drivers involved in traffic jams, as well as to disseminate alert messages about potentially dangerous road stretches, thus allowing drivers to reduce risks and nuisances along their paths.

References

1. Abuelela, M., Olariu, S.: Zipper: A zero-infrastructure peer-to-peer system for vanet. In: International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM). Chania, Crete Island, Greece (2007)
2. Santa, J., Moragon, A., Gomez-Skarmeta, A.F.: Experimental evaluation of a novel vehicular communication paradigm based on cellular networks. In: IEEE Intelligent Vehicles Symposium. Eindhoven, Netherlands (2008)
3. Rybicki, J., Scheuermann, B., Kiess, W., Lochert, C., Fallahi, P., Mauve, M.: Challenge: Peers on wheels—a road to new traffic information systems. In: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (2007)
4. Picone, M., Amoretti, M., Zanichelli, F.: Proactive neighbor localization based on distributed geographic table. In: 8th ACM-SIGMM International Conference on Advances in Mobile Computing and Multimedia (MoMM 2010). Paris, France (2010)
5. Picone, M., Amoretti, M., Zanichelli, F.: Proactive neighbor localization based on distributed geographic table. *Int. J. Pervasive Comput. Commun.* **7**(3), 240–263 (2011)
6. Amoretti, M.: A survey of peer-to-peer overlay schemes: Effectiveness, efficiency and security. *Recent Pat. Comput. Sci.* **2**(3), 195–213 (2009)
7. Aberer, K., Alima, L., Ghodsi, A., Girdzijauskas, S., Haridi, S., Hauswirth, M.: The essence of p2p: A reference architecture for overlay networks. In: 5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005). Konstanz, Germany (2005)
8. Granovetter, M.: The strength of weak ties. *Am. J. Sociol.* **78**(6), 1360–1380 (1973)
9. Maymounkov, P., Mazières, D.: Kademlia: A peer-to-peer information system based on the xor metric. In: First International Workshop on Peer-to-peer Systems (2002)
10. eMule Team: eMule project homepage. <http://www.emule-project.net>
11. Wolinsky, D.I., St Juste, P., Boykin, P.O., Figueiredo, R.: Addressing the p2p bootstrap problem for small overlay networks. In: IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), pp. 1–10. IEEE (2010)
12. Fewell, M.: Area of common overlap of three circles. <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA463920> (2006)

13. Hartenstein, H., Laberteaux, K.: VANET Vehicular Applications and Inter-Networking Technologies. Wiley, New York (2010)
14. Harri, J., Filali, F., Bonnet, C.: A framework for mobility models generation and its application to inter-vehicular networks. In: 3rd IEEE International Workshop on Mobility Management and Wireless Access (MobiWac05). Cologne, Germany (2005)
15. Fiore, M., Harri, J., Filali, F., Bonnet, C.: Vehicular mobility simulation with vanetmobisim. *Simulation* **87**(4), 275–300 (2009)
16. Zhou, B., Xu, K., Gerla, M.: Group and swarm mobility models for ad hoc network scenarios using virtual tracks. In: IEEE Military Communications Conference (MILCOM). Monterey, California, USA (2004)
17. Seskar, I., Marie, S., Holtzman, J., Wasserman, J.: Rate of location area updates in cellular systems. In: IEEE Vehicular Technology Conference (VTC'92). Denver, Colorado, USA (1992)
18. Krauss, S., Wagner, P., Gawron, C.: Metastable states in a microscopic model of traffic flow. *Phys. Rev. E* **55**(1), 55–97 (1997)
19. Krauss, S., Wagner, P., Gawron, C.: Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **62**(2), 1805–1824 (2000)
20. Gustafsson, E., Jonsson, A.: Always best connected. *Phy. Rev. E* **10**(1), 49–55 (2003)
21. Olivera, J., Cortazar, I., Pinart, C., Los Santos, A., Lequerica, I.: Vanba: A simple handover mechanism for transparent, always-on v2v communications. In: 69th IEEE Vehicular Technology Conference (VTC2009-Spring). Barcelona, Spain (2009)
22. Yan, Z., Zhou, H., Zhang, H., Zhang, S.: Speed-based probability-driven seamless handover scheme between WLAN and UMTS. In: 4th International Conference on Mobile Ad Hoc and Sensor Networks. Wuhan, China (2008)
23. Kwak, D., Mo, J., Kang, M.: Investigation of handoffs for IEEE 802.11 networks in vehicular environment. In: 1st International Conference on Ubiquitous and Future Networks (ICUFN). Hong Kong (2009)
24. Esposito, F., Vegni, A., Matta, I., Neri, A.: On modeling speed-based vertical handovers in vehicular networks. In: IEEE Global Telecommunications Conference (GLOBECOM'10). Miami, Florida, USA (2010)
25. Balasubramaniam, S., Indulska, J.: Vertical handover supporting pervasive computing in future wireless networks. *Comput. Commun. J.* **27**(8), 708–719 (2003)
26. Gershenson, C., Heylighen, F.: How can we think the complex? In: Richardson, K. (ed.) *Managing Organizational Complexity: Philosophy, Theory and Application*, pp. 47–71. Information Age Publishing, Charlotte (2005)
27. Zeigler, B.P., Praehofer, H., Kim, T.G.: *Theory of Modeling and Simulation*, 2nd Edn. Academic Press, Massachusetts (2000)
28. Wainer, G.: Cd++: A toolkit to develop DEVS models. *Softw.: Pract. Experience* **32**(13), 1–46 (2002)
29. Varga, A., Hornig, R.: An overview of the OMNeT++ simulation environment. In: 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008). Marseille, France (2008)
30. Amoretti, M., Agosti, M., Zanichelli, F.: Deus: A discrete event universal simulator. In: 2nd ICST/ACM International Conference on Simulation Tools and Techniques (SIMUTools 2009). Rome, Italy (2009)
31. Baldo, N., Requena-Esteso, M., Nin-Guerreo, J., Miozzo, M.: A new model for the simulation of the LTE-EPC data plane. In: 5th ICST/ACM International Conference on Simulation Tools and Techniques (SIMUTools 2009). Sirmione, Italy (2012)
32. Papoulis, A.: *Probability, Random Variables, and Stochastic Processes*. Mcgraw-Hill College (1991)
33. Lee, U., Gerla, M.: A survey of urban vehicular sensing platforms. *Comput. Netw.* **54**(4), 527–544 (2010)
34. Gerla, M., Kleinrock, L.: Vehicular networks and the future of the mobile internet. *Comput. Netw.* **55**(2), 457–469 (2011)

35. Lee, U., Magistretti, E., Zhou, B., Gerla, M., Bellavista, P., Corradi, A.: Mobeyes: Smart mobs for urban monitoring with vehicular sensor networks. *IEEE Wireless Commun.* **13**(5), 52–57 (2006)
36. Jiang, D., Taliwal, V., Meier, A., Holfelder, W., Herrtwich, R.: Design of 5.9 Ghz DSRC-based vehicular safety communication. *IEEE Wireless Commun.* **13**(5), 36–43 (2006)
37. Han, M., Moon, S., Lee, Y., Jang, K., Lee, D.: Evaluation of VoIP quality over WiBro. In: *Passive and Active Measurement Conference (PAM)*. Cleveland, Ohio, USA (2008)
38. Hadaller, D., Keshav, S., Brecht, T., Agarwal, S.: Vehicular opportunistic communication under the microscope. In: *International Conference on Mobile Systems, Applications and Services (MobiSys)*. San Juan, Puerto Rico (2007)
39. Qureshi, A., Carlisle, J., Guttag, J.: Tavarua: Video streaming with wwan striping. In: *14th Annual ACM International Conference on Multimedia*. Santa Barbara, California, USA (2006)
40. EHull, B., Bychkovsky, V., Zhang, Y., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., Madden, S.: Cartel: A distributed mobile sensor computing system. In: *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Boulder, Colorado, USA (2006)
41. Rybicki, J., Scheuermann, B., Koegel, M., Mauve, M.: Peertis: a peer-to-peer traffic information system. In: *Proceedings of the Sixth ACM International Workshop on Vehicular InterNetworking* (2009)
42. Team, J.: JXTA technology: Creating connected communities. Tech. rep, Sun Microsystems (2004)
43. Mohan, P., Padmanabhan, V., Ramjee, R.: Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In: *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Raleigh, North Carolina, USA (2008)
44. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., Srivastava, M.: Participatory sensing. In: *WSW'06*. Boulder, Colorado, USA (2006)
45. Mun, M., Reddy, S., Shilton, K., Yau, N., Boda, P., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R.: Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In: *International Conference on Mobile Systems, Applications and Services (MobiSys)*. Krakow, Poland (2009)
46. Caliskan, M., Graupner, D., Mauve, M.: Decentralized discovery of free parking places. In: *3rd International Workshop on Vehicular Ad Hoc Networks*. Los Angeles, California (2006)
47. Tonguz, O., Wisitpongphan, N., Bai, F., Mudalige, P., Sadekar, V.: Broadcasting in VANET. In: *Workshop on Vehicular Ad Hoc Networks (Move'07)*. Montreal, Canada (2007)
48. Bronsted, J., Kristensen, L.: Specification and performance evaluation of two zone dissemination protocols for vehicular ad-hoc networks. In: *39th Annual Simulation Symposium*. Huntsville, Alabama (2006)
49. Zhao, J., Cao, G.: Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **57**(3), 1910–1922 (2007)
50. Wegener, A., Hellbruck, H., Fischer, S., Schmidt, C., Fekete, S.: Autocast: An adaptive data dissemination protocol for traffic information systems. In: *66th IEEE Vehicular Technology Conference (VTC '07)*. Dublin, Ireland (2007)
51. Shinkawa, T., Terauchi, T., Kitani, T., Shibata, N., Yasumoto, K., Ito, M., Higashino, T.: A technique for information sharing using inter-vehicle communication with message ferrying. In: *IEEE International Conference on Mobile Data Management (MDM '06)*. Nara, Japan (2006)
52. Xu, H., Barth, M.: An adaptive dissemination mechanism form intervehicple communication-based decentralized traffic information system. In: *9th International IEEE Conference on Intelligent Transportation Systems (ITCS '06)*. Toronto, Canada (2006)
53. Fujiki, T., Kirimura, M., Umedu, T., Higashino, T.: Efficient acquisition of local traffic information using inter-vehicle communication with queries. In: *10th International IEEE Conference on Intelligent Transportation Systems (ITCS '07)*. Seattle, Washington, USA (2007)

54. Leontiadis, I., Mascolo, C.: Opportunistic spatio-temporal dissemination system for vehicular networks. In: International Conference on Mobile Systems, Applications and Services (MobiSys). San Juan, Puerto Rico (2007)
55. Distributed Systems Group: ip2 peer project home page. <http://code.google.com/p/sip2peer/> (2013)
56. MjSip Team: Mjsip project official page. <http://www.mjsip.org> (2013)