UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXIV Ciclo

Distributed Algorithms for Location Based Services

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Francesco Zanichelli

Dottorando: *Marco Picone*

Gennaio 2012

*"Non si diventa grandi in qualcosa cercando d'essere grandi,
ma desiderando di fare qualcosa e poi lavorandoci così tanto
da diventare grandi"*

*"A day without laughter is a day wasted."
Charlie Chaplin*

*"Your work is going to fill a large part of your life, and the only
way to be truly satisfied is to do what you believe is great work.
And the only way to do great work is to love what you do."
Steve Jobs*

Summary

List of Figures

List of Tables

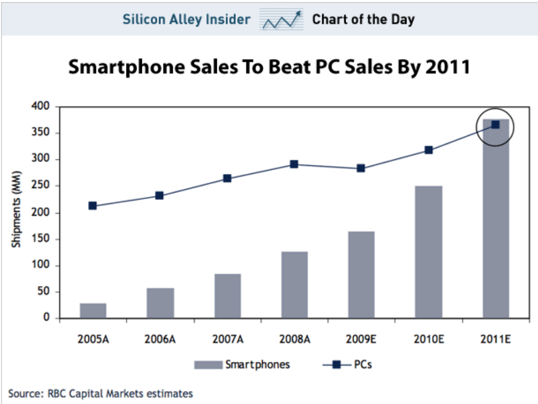
List of Algorithms

Introduction

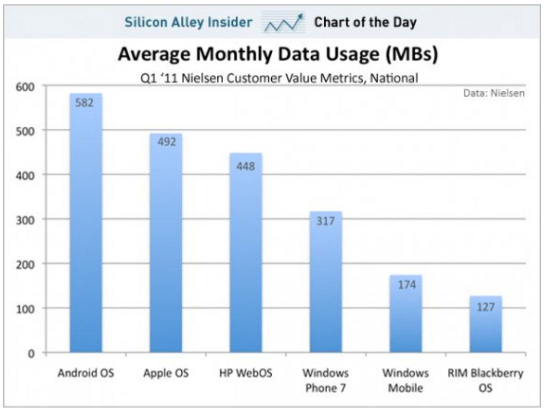
Recent years have seen the relentless market explosion of mobile devices (PDAs, smart-phones, MIDs, PMPs, net-books, etc.), whose ever increasing capabilities make them attractive to an endless number of connected applications and services in business and infotainment domains which can be fully experienced in mobility. The global and extraordinary success of mobile devices is clearly witnessed by the graphs in Fig. 1, where (a) highlights to what extent smartphones are nowadays becoming the commonest computing platform of choice.

Such technological and societal scenarios have been long considered by the well-established research areas of *Mobile* and *Ubiquitous Computing*. The latter in particular is a general term used to describe techniques and technologies that enable people to access computing services anyplace, anytime, and anywhere. Mobile computing has three main facets: mobile communication, mobile hardware, and mobile software [82].

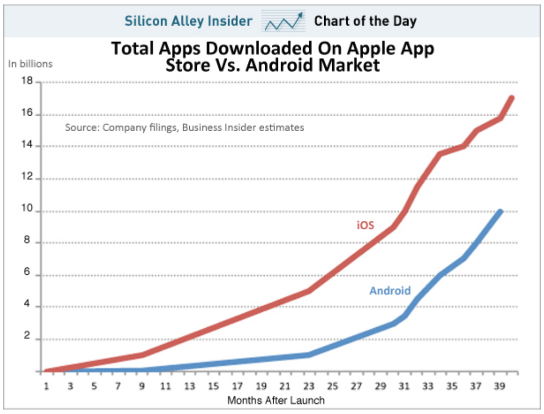
The first aspect addresses communication issues in ad-hoc and infrastructure networks as well as communication properties, protocols, data formats and concrete technologies. Up to a few years ago, broadband systems were only those allowing a high speed data exchange among stationary nodes through transmission channels such as xDSL, satellite, optical fiber or wireless broadband. Today, *mobile broadband* has become a more attractive opportunity for service providers, with an ever growing demand for ubiquitous broadband connectivity as well as performance im-



(a)



(b)



(c)

Figure 1: (a) Comparison between Smartphone and PC Sales by 2011. (b) Average monthly data usage (MBs). (c) Total app downloads on AppleStore and Android Market. (Source Silicon Alley Insider)

provements in mobile network technologies, network footprints and mobile devices. Mobile broadband is also quickly becoming a necessity for a country to remain competitive. For those reasons novel networking technologies with similar or different goals reach every day the mobile market, in the areas of cellular networks (3G/4G), wireless technologies (IEEE 802.11x, WiMax) and ah-hoc or short range communication such as NFC and RFID (Fig. 1 (b) illustrates the average monthly data usage in MBs per user in the U.S. during Q1/2011).

The second facet concerns the hardware. Market-driven huge research and technology efforts are constantly improving device capabilities in terms of computational power and available sensors. Tablets and smartphones of the latest generation are endowed with multi core CPUs, advanced GPUs and gigabytes of memory, thus reshaping the vision and the approach to the category. Modern devices provide also a relevant number of embedded sensors such as cameras, microphones, accelerometers, compass, GPS and have the possibility to connect to external devices through different wired and wireless technologies. All those hardware advancements and the new ubiquitous nature completely changes the target and the application fields of mobile devices making them attractive as (potentially countless) sources of heterogeneous and up to date information.

The third aspect deals with the characteristics and requirements of mobile applications and how software design is changing to follow better the increased capabilities of devices, the requirements of developers and the expectation of users. The mass mobile market has in fact also changed the way applications were traditionally distributed to huge number of users. New software distribution methods have been recently introduced, with the dominance of online application stores (such as AppStore and Android Market), Fig. 1 (c) shows Apple and Google mobile application market trends months after launch. Such virtual marketplaces have solved many problems to developers and common users, allowing easy dissemination and installation of apps, with specific security and update management policies, helping market growth to reach myriads of new end users and target application fields.

All combined these aspects contribute to depict the actual scenario of mobile computing where millions of users at the same time have access to very capable devices to join the network and daily exchange gigantic amount of heterogeneous data. As mentioned earlier, this context becomes more and more attractive for new applications and new research topics. However, at the same time the challenges to be faced go hand in hand with the market growth. One clear example is the research field of distributed computing, where the heterogeneity of involved actors is one of the most important aspects that usually is taken into account by a designer. Nowadays, it is absolutely unrealistic to envision a modern distributed system, *e.g.* a peer-to-peer overlay for live or on-demand streaming, file sharing or data storage, without taking into account the presence and the constraints of mobile devices. This requires to evaluate and address some specific issues to provide an high quality of service for mobile users without affecting the performance of other users. Most notable issues are a reduced connection stability (because of being wireless and moving across different access networks) and limited autonomy (because of being battery-powered) and a different user behavior or application dynamics (background activities, frequent change of context and competition for available resources).

One of the most important research field that have developed along mobile devices evolution and diffusion is related to the challenges of applications based on the geographic location of users and information. Location-Based Services (LBSs) are information or entertainment services where the request, the response and served contents depend on the physical position of the requesting device [83]. LBSs are clearly oriented to mobile devices and users to whom services can be provided and tailored according to their current geographical position. In general, the entity to be localized not necessarily overlaps with the service user.

LBSs can be used in a large variety of contexts, such as health, entertainment, work, personal life, etc., for both common (*e.g.* indoor identification and localization of objects) and domain specific tasks. LBSs include services to identify a location of a person or object, such as discovering the nearest ATM or the whereabouts of a friend or employee. Parcel tracking and vehicle tracking services are thus instances of LBSs which can also include mobile commerce applications taking the form of coupons or advertisements directed at customers depending on their current location. LBS endless field of application include also emerging services such as personalized weather services and location-based games.

Real-time localization services are some of the most challenging and interesting mobile broadband applications in the LBS world. They are gaining more and more importance for a broad range of applications, such as road /highway monitoring, emergency management, social networking, and advertising.

The vast majority of proposed architectures for real-time LBSs are based on a pure centralized or in some cases on a hierarchical infrastructure where one or more central servers have the responsibility to manage all position updates and queries from involved users, related for example to a specific point-of-interest, neighborhood discovery or path planning.

In order to cope with huge numbers of active users at the same time while offering a high quality of service (QoS) usually those solutions require on the service provision side massive computational and network capacities and can be only provided by large companies such as Google and TomTom. In fact, the major technical challenge of centralized systems is how to deal with the high amount of simultaneous updates and queries, as pointed out by Hull et al. [27]; Rybicky [39] - recall that each car is a source of queries and sends its own data (measurement, check-in, location update, etc.) regularly. Owing to the high degree of centralization, servers can thus become bottlenecks or even single point of failures. Centralized approach would hardly scale to the demands of highly variable, massive application contexts, with mobile nodes changing their location very quickly, in a wide and highly populated area. Indeed, search results would be incomplete or outdated with high probability. As proposed by several researchers (see the Related Work section), a partially or fully decentralized approach is able to increase the accuracy of information and the rate at which it can be retrieved by users. Moreover, it may allow to update and publish information directly, with low cost and high scalability. Last but not least, it may simplify the process of joining the virtual community and publish new services.

On other hand, one of the main reasons not to rely on a centralized system for managing location based information (such as traffic information) could be non-technical. It simply does not seem to be desirable to hand the control of this data over one single organization, potentially limiting the access to data collected conjointly by all participants.

Previous considerations motivate the study of a real-time LBS supported by an infrastructure implemented as a network of distributed software mobile entities, with flat or hierarchical organization without relying on a central entity. A typical application field of such LBSs could be a traffic information system spread over a set of mobile nodes moving along city streets, each one collecting and disseminating information about local traffic, and being able to provide aggregated information (*e.g.* statistics) to any remote user that requests it.

Research Problem and Thesis Contribution

This Ph.D. thesis[1] focuses on the problem of defining a new category of decentralized peer-to-peer (P2P) algorithms for location-based services. We aim at defining a P2P overlay where each participant can efficiently retrieve node and resource information (data or services) located near any chosen geographic position. The idea is that the responsibility and the required resources for maintaining information about position of active users are properly distributed among nodes, for which a change in the set of participants causes only a minimal amount of disruption without reducing the quality

[1] This Ph.D fellowship has been funded by a grant from the University and Research Ministry, titled "Fondo Sostegno Giovani", ambito di indagine 8 "Sistemi di telecomunicazione innovativi a larga banda anche con impiego di satelliti per utenze differenziate in materia di sicurezza, prevenzione e intervento in caso di catastrofi naturali"

of provided services.

First of all we argue that, thanks to P2P infrastructural properties, as already demonstrated for several types for distributed applications such as file sharing, media streaming and information dissemination solutions, it is possible to manage geographic information of a huge number of active users at the same time in a distributed network without the control and the assistance of a centralized authority. Secondly, is absolutely mandatory to consider the heterogeneity of user devices not only in terms of bandwidth and computational capabilities but also in terms of available embedded or external sensors and in particular mobility.

We claim that a peer-to-peer solution is an appealing approach to build a base layer for LBS as it will demonstrated later on in the thesis. A P2P approach is high scalable with a relevant number of concurrently active nodes, it is robust to multiple failures, is a low cost solution where each user can contribute to the platform with resource and services and in our scenario is natively ready also to support next generation ad-hoc WiFi and near field communication (NFC) between mobile devices, without significant changes in the protocols.

In this thesis we will assess the validity of the proposed model through a formal analysis of the routing protocol and a detailed simulative investigation of the designed overlay. We will depict a complete picture of involved parameters, how they affect the performance and how they can be configured to adapt the protocol to the requirements of several location based applications. Furthermore we will present two application scenarios where the designed protocol has been simulated and evaluated, as well as the first prototype of a real implementation of the overlay using both traditional PC nodes and Android mobile devices.

The main contributions of this thesis can be summarized as follows:

- An analysis on how existing location-based algorithms manage user geo-spatial queries and disseminate the information. We will compare existing central and decentralized approaches to understand which are the main drawbacks and how they can be overcame to design a new overlay, considering the heterogeneity of the network and of the involved devices.

- The proposal of a novel peer-to-peer algorithm called *Distributed Geographic Table* (DGT) based on the geographic location of involved users and information. The DGT algorithm allows to obtain a system where overlay neighbors are at the same the the real geographic neighbors of a peer without the need to route additional messages.

- A performance and feasibility evaluation of our approach through a formal analysis of the main routing and discovery protocol showing how it is able to retrieve updated information about the neighborhood of a generic geographic point in a limited number of steps. The evaluation has been enriched with an intensive and realistic simulation phase with an accurate modeling of peer mobility aiming to study the impact of different parameters involved in the algorithm and how they can influence the performance of the protocol.

- An investigation of the performance of the Distributed Geographic Table in two different appealing scenarios. Firstly, we take advantage of the designed overlay to build a distributed traffic information system (TIS) for the dissemination of traffic alert messages. In such a system users can participate using their smartphone to send and receive real time information about traffic conditions or potentially dangerous situations. The second application takes advantage of a joint DGT and network coding approach for large scale information management in smart cities.

- An implementation of the first DGT prototype to evaluate the performance of the protocol in a real smartphone based vehicular network. The algorithm and the overlay have been implemented on traditional PC and on the Android platform using the P2P middleware called sip2peer. The prototype has been tested both on the PlanetLab testbed with several nodes and on the field with a small number of smartphones and vehicles.

Thesis Outline

The thesis is organized as follows:

- Chapter 1: The chapter describes and analyzes the literature of centralized and distributed algorithms and systems related to LBSs. Among several applications and research projects we selected the most relevant and innovative approaches to illustrate current state of the art and motivate the solution proposed in this thesis.

- Chapter 2: In this chapter we describe and define our decentralized algorithm LBS providing a formal analysis of the discovery procedure and an extensive simulation analysis in different scenarios and configuration. We also present the modeling theory representing the base layer that allows us to implement a realistic simulation in terms of user mobility, network and communication modeling.

- Chapter 3: In the chapter we prove the validity of the designed approach as we evaluated the DGT overlay in two high dynamically scenarios: a smartphone based Traffic Information System (TIS) for the opportunistic dissemination of warning messages to drivers and a SmartCity architecture for the management of large scale information from raw and aggregated data sources.

- Chapter 4: Finally, this chapter concludes this thesis and provides some general discussion and outline further work.

Chapter 1

Related Work

Real-time localization services are some of the most challenging and interesting mobile broadband applications in the LBS world. They are gaining more and more importance for a broad range of applications, such as road/highway monitoring, emergency management, social networking, and advertising. In this chapter we analyze the most interesting approaches and algorithms grouped according to the type of their architecture.

1.1 Centralized Approaches

Traditionally proposed architectures are based on a centralized approach where one or more central server has the responsibility to manage all position updates and queries from involved users related for example to a specific point-of-interest, neighborhood discovery or path planning. In order to manage a huge number of active users at the same time with a high quality of service (QoS), usually those solutions require on the server side a relevant computational power and are provided by big companies such as Google and TomTom.

Google Latitude [4] has been introduced by Google in 2009 and is a location-aware web/mobile application that allows a mobile phone user to share with a certain group of people (real and social friend) his/her current location (Fig. 1.1). Via their

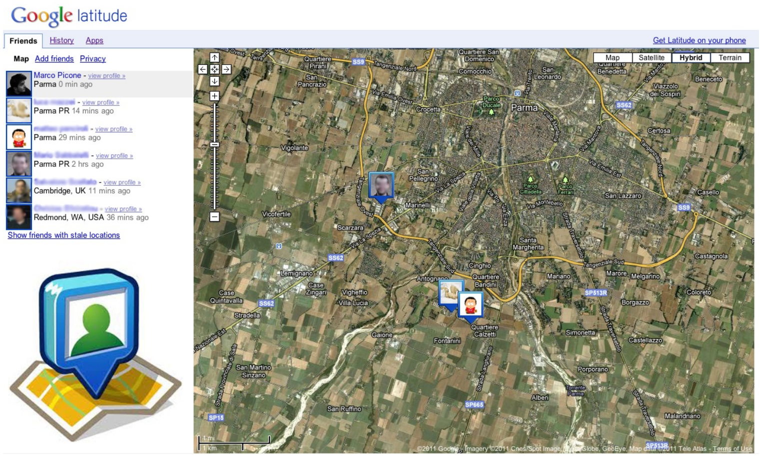


Figure 1.1: Google Latitude web application with friend list and map view.

own Google Account, the user's cell phone location is mapped on Google Maps. The user can control the accuracy and details of what each of the other users can see. An exact location can be allowed, or it can be limited to identifying the city only. For privacy, it can also be turned off by the user, or a location can be manually entered. Users can set specific privacy options, and may only see the location of those friends who have decided to share their location with them. Recently Google added the possibility for the user to share its check-in during the day with latitude friends, merging the functionalities of Google Places application.

Another example of centralized location based architecture is TomTom HD Traffic. HD Traffic is TomTom's real-time traffic service that tries to give an accurate and up-to-date traffic information. HD Traffic is part of TomTom's LIVE Services that deliver essential information services to drivers helping them to save time, money and fuel. In order to be able to provide such accurate real-time information on all major and secondary roads, TomTom's patented HD Traffic technology uses, above all, traffic data generated by the movement patterns of mobile phones inside vehicles collected anonymously from the mobile carrier (for example Vodafone Italy) networks. This is then combined with anonymous data from TomTom devices as well as other

Figure 1.2: TomTom HD Traffic Idea and Web Interface

traditional sources of traffic information, to provide the world's most advanced traffic information service. The traffic information is relayed in real-time and securely to TomTom devices, thanks to Vodafone Italy's patented Machine to Machine (M2M) solutions [12] and includes a SIM card with a GPRS connection that have been built into the navigation device. Processed information and evaluation results such as traffic status, accident and road monitoring are then available on TomTom devices or through a web interface (Fig.1.2).

Another example of centralized location base system is CarTel. It is a mobile sensor computing system designed to collect, process, deliver, and visualize data from sensors located on mobile units such as automobiles. A CarTel node is a mobile embedded computer coupled to a set of sensors. Each node in the system gathers and locally processes sensor readings, before delivering them to a central portal, where data are stored in a database for further analysis and visualization. CarTel provides a simple query-oriented programming interface, handles large amounts of heterogeneous data from sensors, and handles intermittent and variable network connectivity. CarTel nodes rely primarily on opportunistic wireless (e.g., Wi-Fi, Bluetooth) connectivity

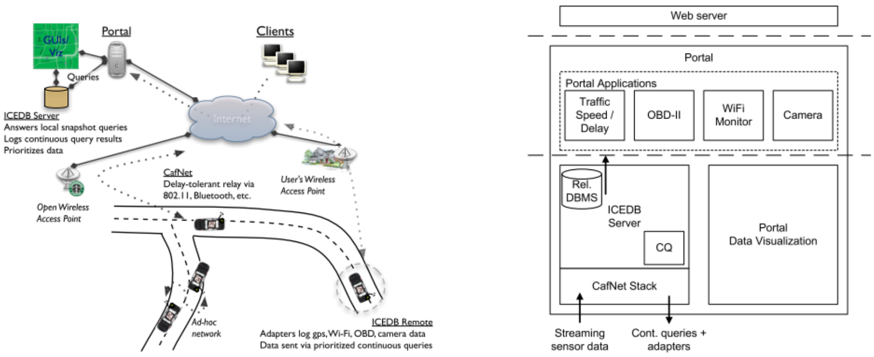


Figure 1.3: CarTel Architecture

to the Internet, or to "data mules" such as other CarTel nodes, mobile phone flash memories, or USB keys to communicate with the portal. CarTel applications run on the portal, using a delay-tolerant continuous query processor, ICEDB, to specify how the mobile nodes should summarize, filter, and dynamically prioritize data. Figure 1.3 illustrate the system architecture showing the different components of the platform. Cars collect data as they drive, and log them to their local ICEDB databases. As connectivity becomes available, data on cars is delivered to the portal, where users can browse and query it via a visualization interface and local snapshot queries.

Provided examples of centralized solutions are only a few number of applications among a huge amount of existing solutions. They allow to depict an idea about problems associated to location based solutions, the amount and heterogeneity of information that could be potentially involved, and the number of simultaneously active users that participate in such a system.

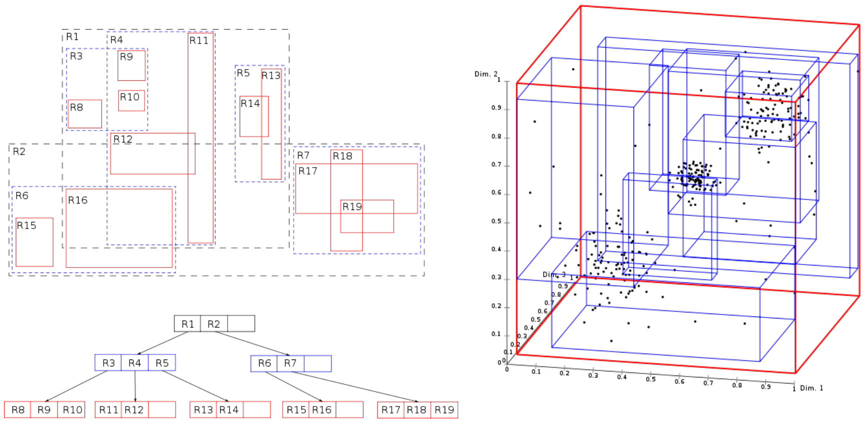
1.2 Distributed Approaches

The research during the last decades due to the progressive improvements of Internet connection and in particular of mobile devices capabilities, moved to study and analyze distributed algorithms and overlays for different purposes such as file sharing, social application, live and on demand streaming and naturally decentralized geolocation services.

Distributed localization is a clear example of geocollaboration service, usually implemented by recursively dividing the 2D space into smaller areas in order to assign responsibilities for region of space to peers. Instead of employing a number of centralized servers (either dedicated or selected among participating nodes) to carry the load for the entire network, every node in the network shares the load of indexing and searching data that refers to its area. The idea of hierarchical partitioning comes from the indexing of data structures for multidimensional data-sets such as R-tree [11] that is widely used in centralized databases. R-trees are tree data structures used for spatial access methods, i.e., for indexing multi-dimensional information such as geographical coordinates, rectangles or polygons. The proposed approach has found significant use in both research and real-world application. It has been used for example to store spatial objects such as restaurant locations or the polygons that typical maps are made of: streets, buildings, outlines of lakes, coastlines, etc., and then quickly find answers to queries such as "*Find all museums within 2 km of my current location*", "*retrieve all road segments within 2 km of my location*" or "*find the nearest gas station*".

The key idea of the data structure is to group nearby objects and represent them with their minimum bounding rectangle in the next higher level of the tree. Since all objects lie within this bounding rectangle, a query that does not intersect the bounding rectangle can not intersect any of the contained objects. At the leaf level, each rectangle describes a single object; at higher levels it describes the aggregation of an increasing number of objects. This can also be seen as an increasingly coarse approximation of the data set. Figure 1.4 (a) and (b) show R-Tree examples in the 2D and 3D space.

An overlay structure allows for routing queries within the system. This means that each time it is necessary to know which peers are located in a certain area, a number of lookup queries must be sent. Examples of general-purpose hierarchical peer-to-peer schemes supporting *geocollaboration* services are HZSearch [7], DPTree [6], DiST [8]. These and other works [9], [10], [3] propose strategies for supporting complex



(a) (b)

Figure 1.4: (a) Simple example of an R-tree for 2D rectangles. (b) Visualization of an R*-tree for 3D cubes.

queries over multi-dimensional data, such as "select five available buildings closest to the airport".

The specific problem of geographic localization is addressed for example by Globase.KOM (Geographical LOcation BAsed SEarch) [1], which adopts a tree-based P2P overlay enhanced with interconnections.Their main focus is to enable search over all peers in some defined geographical area. The area can be circular or rectangular. A peer can search for a node with some particular location, or for the geographically closest peer. Together with the information about its geographical location, a peer can publish any other data describing the service it offers (e.g. a video stream from a webcam), the object it represents (e.g. restaurant, police station, sightseeing, gasoline station), or some additional information (e.g. menu, prices, opening hours). For example, users can find the closest gasoline station or can find all restaurants in some area and see their menu or video streams from webcams.

The Globase.KOM scheme is based on supernodes, *i.e.* powerful nodes, with best network connectivity, that tend to stay online for a long time. Supernodes are respon-

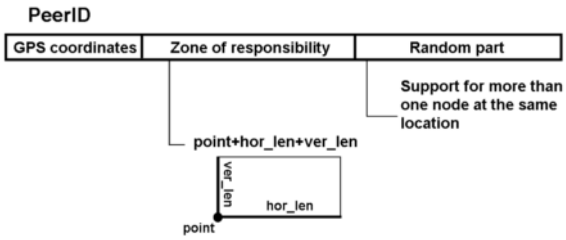
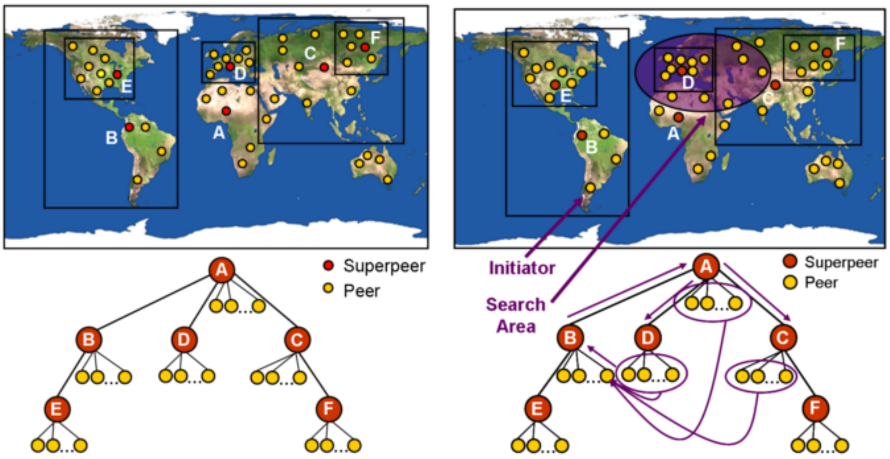


Figure 1.5: Globase.KOM Structure of PeerID

sible for indexing all nodes/services in one clearly defined geographical area. Other nodes in the network simply offer and consume services without having additional responsibilities. The idea is that the world projection is divided into disjoint, non-overlapping zones. Each zone is assigned to a supernode (located inside the zone itself) that has to collect, store and maintain the overlay/underlay contact addresses of all nodes in that zone (Fig. 1.6(a)).

Each superpeer maintains the contact addresses of the peers inside of its zone, excluding the inner zones, superpeers responsible for inner zones (i.e. child nodes), its parent in the tree, the root superpeer, and interconnected superpeers . Each peer maintains the following contact addresses: the parent superpeer, the root superpeer, an interconnection list, and a cache list of already contacted peers. Peers/Superpeers are identified by their unique ID (Fig. 1.5) that contains: the GPS coordinate of the node, if it is a supernode, the zone it is responsible for, and a random part in order to support the existence of more than one peer at the same location.

Active peers in Globase.KOM can perform three main location based operations such as area searches, lookups or finding the geographically closest node. Area search is performed using the SEARCH message which includes a description of the geographical area (center and radius) plus metadata describing the targeted service/object. When a superpeer receives a SEARCH query from one of its peers, it calculates the searched ellipse onto the map projection. Next, it checks if that ellipse intersects the zone it is responsible for.



(a) (b)

Figure 1.6: (a) Globase.KOM division of world projection in multiple non-overlapping zones. (b) Example of Globase.KOM Area Search procedure.

Figure 1.6(b) illustrates an example of area search. A peer in the zone of superpeer B sends a SEARCH message containing a description of the marked zone. As the zone does not intersect the zone superpeer B is responsible for, the SEARCH message will be forwarded to the superpeer A. In the end, superpeers A, C, and D will reply with the list of the matching results.

The lookup operation is used to determine the underlay address (IP address and port) of a peer from its geographical location. Each superpeer knows the IDs/locations of all nodes it is responsible for. Therefore, a lookup operation basically means routing the LOOKUP message to the superpeer responsible for the peer with the given location.

When a peer wants to find the closest peer, it first calculates the closest border of the zone it belongs to. This is possible by using the ID of the parent superpeer, which contains a vector representation of the zone. Then, the peer sends a FIND CLOSEST message to its parent superpeer, containing the calculated distance to the closest border of the zone.

Another architecture, called GeoP2P [2], still performs a hierarchical partitioning of the 2D geographic space, but adopts a fully decentralized peer-to-peer overlay scheme, with overlay maintenance and query routing performed without super or special peers. The system consists of large number of peers, distributed across a 2-dimensional space with rectangular boundary. Each peer resides in and a point in the 2 dimensional space and responsible for providing information relevant to that point. A peer can be a data collection sensor such as a surveillance camera or a database regarding a particular object related to the point such as a hotel or gas station. The data stored in each peer is updated independently. Also, any peer can be interested in any region in the space and launch a query. The purpose of the overlay network is to route the query to all relevant peers.

The universe is hierarchically divided into zones. At the top level of the hierarchy, the zone representing the universe is divided into a number of sub-zones, each of the sub-zones being further divided into sub-sub-zones at the next level of the hierarchy, and so on. Thus the zones can be conceptually organized into a tree, where the root of the tree rep- resents the universe and each tree-node represents a zone.

Figure 1.7(a) illustrates an example division of the universe and the corresponding tree representation is shown in Figure 1.7(b).

Each peer maintains a routing table that lists all the other peers it knows. To resolve a query about any region in the universe, a peer tries to find a peer that belongs to the leaf zones intersecting the query region. To do that, each peer needs to have some structured knowledge to cover the globe, such that for any zone, it either knows all the peers belonging to that zone, or at least knows some peer that knows more about that zone. Any query can thus be either resolved or forwarded to a peer that has better knowledge of the queried region.

The routing table is organized in d rows, one for each level of hierarchy from1 to d. Each row maintains information regarding $k-1$ sibling zones of that level, plus some in- formation for the self-zone. For each sibling zone, the table need to maintain the network address of one (or more) contact peer, rectangular boundary (coordinates of bottom-left and top-right corner) of the zone. Siblings can be orga-

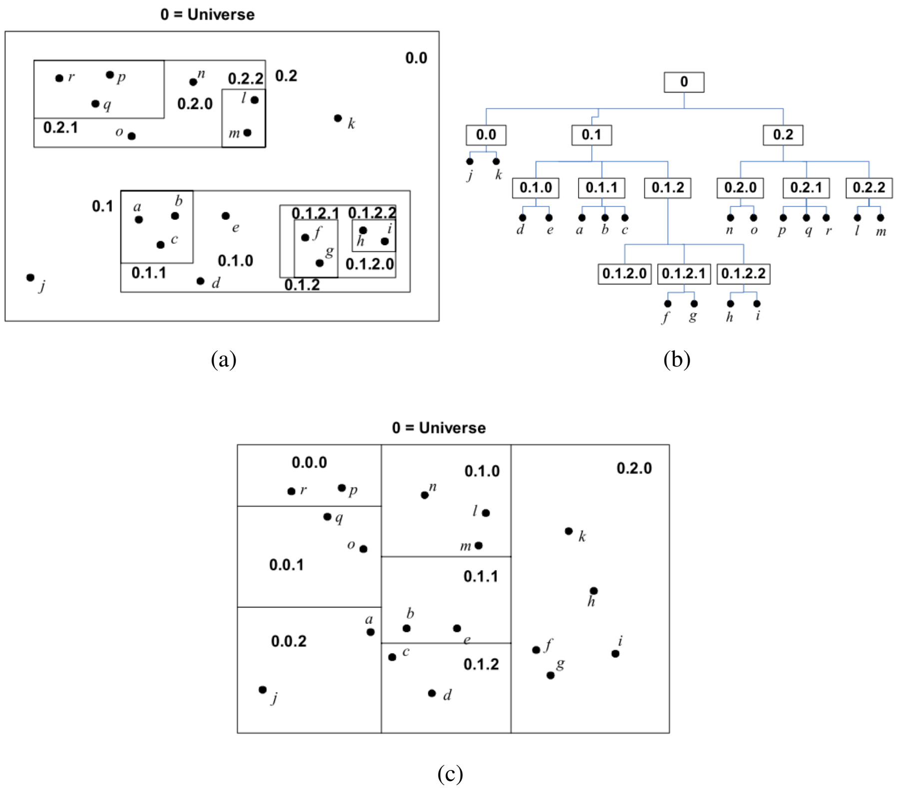
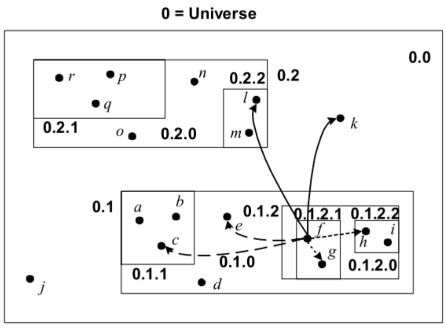


Figure 1.7: (a) Zoning by clustering. (b) Zoning Hierachy. (c) Zoning by splitting.

nized into columns based on the segment of the zone id that identifies the branching at that level. For the self-zone, only the zone boundary need to be maintained, and it can be stored in the corresponding column based the id of the self-zone. Level d, stores the information regarding the leaf zone and here the siblings are individual peers instead of zones. So, coordinates of the peers are stored here instead of rectangular boundaries.1.8(b) shows an example routing table of a peer. The same overlay neighborhood is illustrated in Figure 1.8(a)

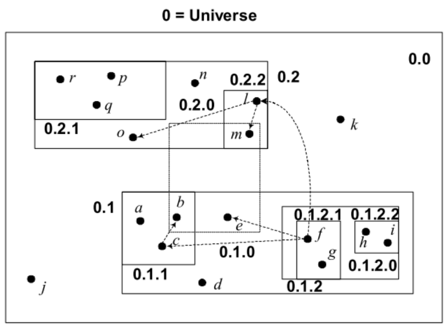
The main drawback of the hierarchical approach is that peers representing higher level regions may become bottlenecks for query routing, and possible points of failure for the whole system. Moreover, none of the state-of-art solutions has been demonstrated to work in presence of mobile peers.



(a)

level		0	1	2	
1	boundary	0.0	0.1	0.2	← boundary
	contact	*k*	self	*l*	← contact
2	boundary	0.1.0	0.1.1	0.1.2	
	contact	*e*	*c*	self	
3	boundary	0.1.2.0	0.1.2.1	0.1.2.2	
	contact		self	*h*	
4	boundary	0.1.2.1.0	0.1.2.1.1		
	contact	*g*	self		

(b)



(c)

Figure 1.8: (a) Overlay neighbors of peer f. (b) Routing table of peer f. (c) Routing path of a range query.

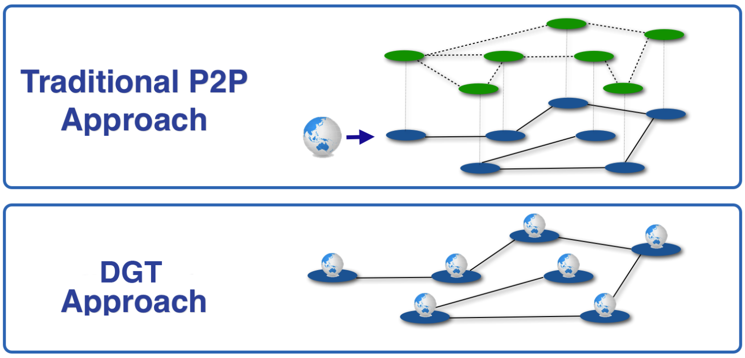
Chapter 2

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 [14]. 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



(a)

Figure 2.1: Comparison between traditional P2P approaches and the Distributed Geographic Table

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. 2.1).

In the following chapter, we present the DGT approach, using the P2P system notation introduced by Aberer *et al.* [13].

2.1 Conceptual Model and Neighborhood

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

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

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):

$$d : \mathscr{W} \times \mathscr{W} \rightarrow \mathbb{R}. \tag{2.1}$$

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

$$\mathscr{N} : \mathscr{W} \times \mathscr{A} \rightarrow 2^{\mathscr{P}} \tag{2.2}$$

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

$$\mathscr{N} = \{p \in \mathscr{P} | w_p \in A_t\} \quad t \in \mathscr{W}, A_t \in \mathscr{A} \tag{2.3}$$

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

$$C_t = \{w \in \mathscr{W} | d(w,t) \leq c_r\} \quad t \in \mathscr{W} \tag{2.4}$$

$$\mathscr{N} = \{p \in \mathscr{P} | w_p \in C_t\} \quad t \in \mathscr{W}, a \in \mathscr{A}. \tag{2.5}$$

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

2.2 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 $w \in \mathscr{W}$.

Routing is a distributed process implemented as asynchronous message passing. By executing the $route(p,w,a)$ operation, a peer forwards to another peer $p \in \mathscr{P}$ a request for the list of nodes that peer p knows to be located in the region $a \in \mathscr{A}$, whose center is $w \in \mathscr{W}$. Thus, a routing strategy can be described by a possibly non-deterministic function:

$$\mathscr{R} : \mathscr{P} \times \mathscr{W} \times \mathscr{A} \to 2^{\mathscr{P}} \tag{2.6}$$

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

The routing process is based on the evaluation of the region of interest centered in the target position. The idea is that each peer involved in the routing process selects, among its known neighbors, those that presumably know a large number of peers located inside or close to the chosen area centered in the target point. If a contacted node cannot find a match for the request, it does return a list of closest nodes, taken from its routing table. This procedure can be used both to maintain the peer's local neighborhood $\mathscr{N}$ 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 [22], 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 [35], used, for example, in recent versions of the eMule client (as an alternative to the traditional eDonkey protocol) [50]. 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.

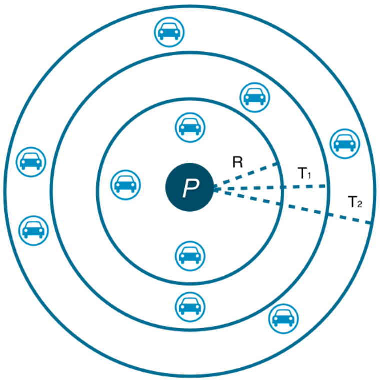
The main goal of the DGT protocol is to build and maintain an overlay where each node knows all the active nodes that are available in a geographic region, in order to implement and provide specific applications and services. An example of application based on such a protocol may be a city monitoring 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.

2.3 Data Structures

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. 2.2). 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;



(a)

Figure 2.2: GeoBucket data structure.

- *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.

2.4 Network Join

When a new peer wants to enter the network, it sends a join request, together with its *GP*, to a *bootstrap node*, that returns a list of 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.

2.5 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.

Node n searches in the GB associated to the requested GP. The final objective of the lookup (summarized in Algorithm 2.1) 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 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 $\mathscr{C}_i = \{n_{1i}, .., n_{\alpha i}\}_{i=1}^{K}$, where i is an integer index. The initiator sends parallel *FIND_NODES* requests, using its *GP* as target, to the α peers in $\mathscr{C}_i$. 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.

Algorithm 2.1 Periodic Lookup Algorithm

```
i ← 0
get α nodes from geo-buckets (nearest to GP): C_i = {n_1i, .., n_αi}
repeat
    j ← 1
    while j ≤ α do
        if n_ji not yet queried then
            n_ji.FIND_NODES(GP)
        end if
        j ← j + 1
    end while
    get α nodes (nearest to GP) from the αβ results: C_{i+1}
    i ← i + 1
until C_{i+1} == C_i
f ← i
get K nodes (nearest to GP) from geo-buckets, not already in C_f
j ← 1
while j ≤ K do
    if n_ji not yet queried then
        n_ji.FIND_NODES(GP)
    end if
    j ← j + 1
end while
```

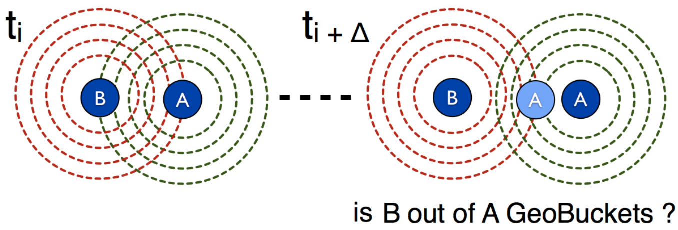


Figure 2.3: DGT Position Update

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 $T_{\max}$.

2.6 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. 2.4). 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. 2.3).

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

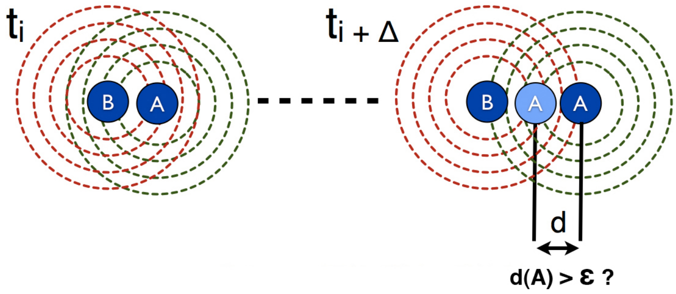


Figure 2.4: DGT neighbor remove process.

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.

2.7 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.

2.7.1 Analytical Performance Evaluation Framework

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 distribution[1] 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 Section 3.1.8, 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. 2.5, 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

[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.

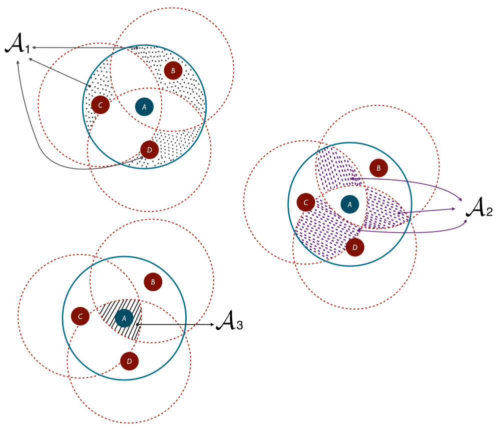


Figure 2.5: Intersection regions (with corresponding areas $\{A_j\}$) between $\alpha = 3$ overlapping circular areas of interest.

and j contacted peers. In Fig. 2.5, the areas $\{A_1, A_2, A_3\}$ are indicated. These areas will be computed using the MatLab library available at [45]. Explicit expressions (not shown here for the sake of conciseness) can be derived according to the analysis in [18].

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

$$\overline{n}(s) = \begin{cases} 0 & s = 0 \\ l_0 & s = 1 \\ \overline{n}(s-1) + \sum_{j=1}^{\alpha} \overline{d}_j\left(\overline{n}(s-1)\right) & s \geq 2 \end{cases} \tag{2.7}$$

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

$$\overline{d}_j\left(\overline{n}(s-1)\right) = \rho \cdot A_j \cdot (1-x) \cdot b_j\left(\overline{n}(s-1)\right). \tag{2.8}$$

In (2.9), $b_j\left(\overline{n}(s-1)\right)$ 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:

$$b_j\left(\overline{n}(s-1)\right) = \left[1 - \frac{\overline{n}(s-1)}{\overline{N}_D^{(R)}}\right]^j. \tag{2.9}$$

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

$$\overline{n}(s) = \overline{n}(s-1) + \sum_{j=1}^{\alpha} \rho \cdot A_j \cdot (1-x) \cdot \left[1 - \frac{\overline{n}(s-1)}{\overline{N}_D^{(R)}}\right]^j. \tag{2.10}$$

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

In Fig. 2.6, the performance results predicted by the analytical model proposed above are compared with simulation results (obtained by means of DEUS, a tool that we describe in the following sub-section), considering scenarios with (a) 500 peers and (b) 1000 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 $\overline{n}(1) = 10$ 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. 2.7 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. 2.6. By observing the results in Fig. 2.7 and Fig. 2.6, it can be concluded that a small number of discovery steps is sufficient, regardless of the value of α, to significantly reduce the PMN.

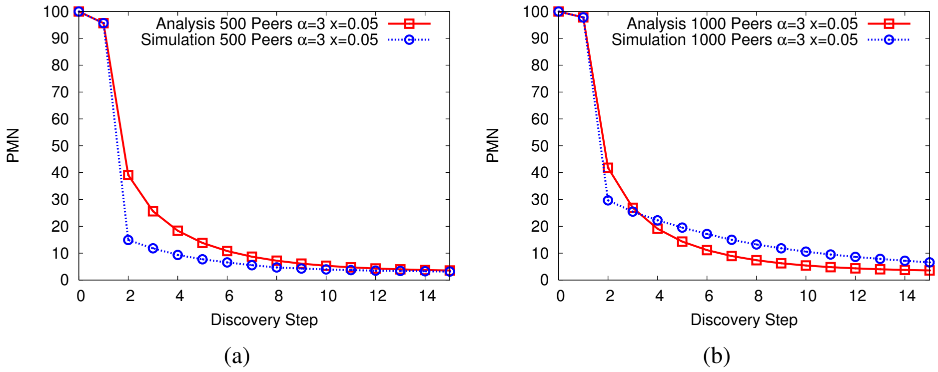


Figure 2.6: PMN as a function of the discovery step, considering (a) 500 peers and (b) 1000 peers. In all cases, $\alpha = 3$.

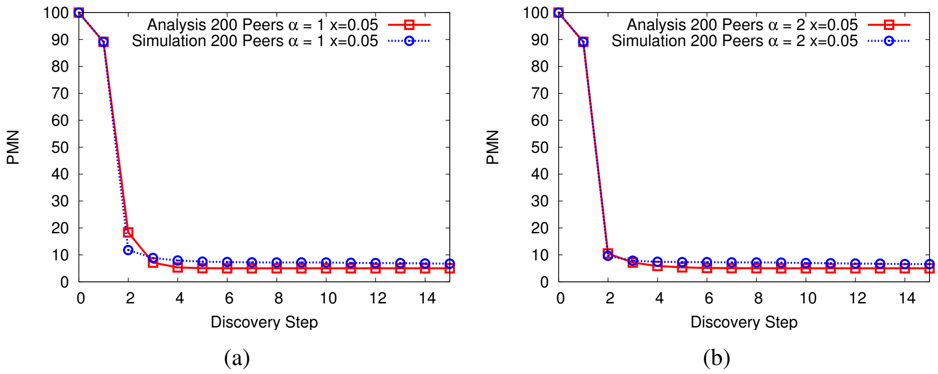


Figure 2.7: PMN as a function of the discovery step, considering (a) $\alpha = 1$ and (b) $\alpha = 2$. In all cases, 200 peers.

2.7.2 Definition of a suitable Mobility Model

Mobility Models (MM) 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.

In our work we focus 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 2.8 illustrates five major MM categories that can be summarized according to [26] in:

- **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 threes different classes for flow models:

  - **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,

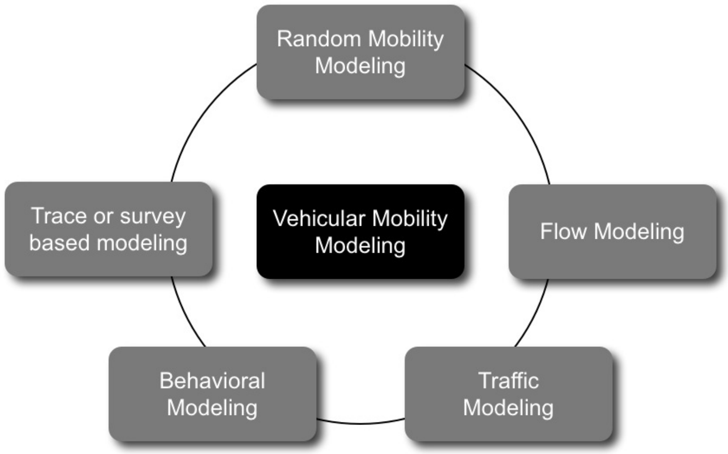


Figure 2.8: Classification of vehicular mobility modeling approaches

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 Figure 2.8 and the concept and idea proposed

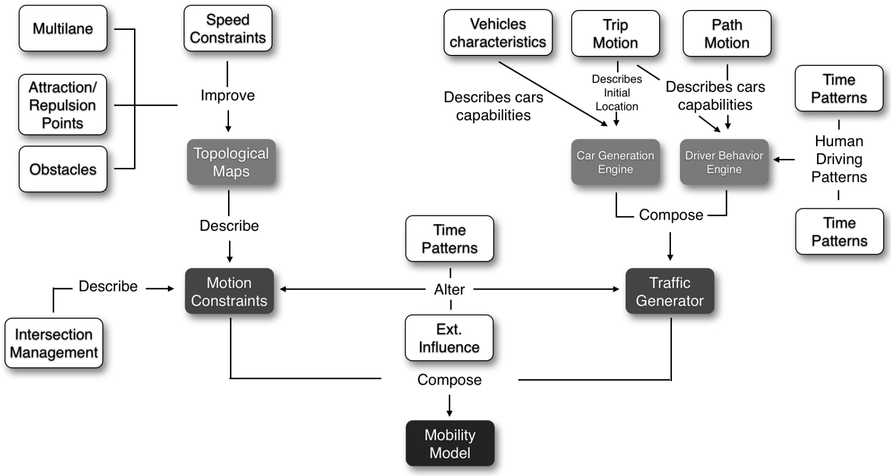


Figure 2.9: Concept map of for the design of vehicular mobility models.

in [25] and [19], mobility models intended to generate realistic vehicular motion patterns should include the following features.

- **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.
- **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, decel-

eration, 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.

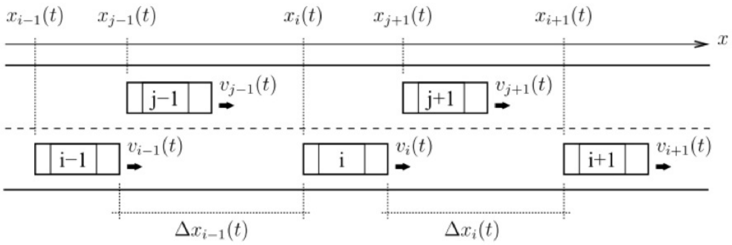


Figure 2.10: Vehicular Mobility Model Notation

- **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 2.10 illustrates 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.

Our model partially follows the approach of Zhou *et al.* [49] 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. 2.11). 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

Symbol	**Definition**
Δt	time step in $[s]$
a	maximum acceleration in $[m/s^2]$
b	maximum deceleration in $[m/s^2]$
L	car length in $[m]$
v^{min}	minimum velocity in $[m/s]$
v^{max}	maximum velocity in $[m/s]$
v^{des}	desired or targeted velocity in $[m/s]$
v^{min}	minimum velocity in $[m/s]$
θ^{max}	maximum heading in $[rad]$
θ^{min}	minimum heading in $[rad]$
T	safe time headway $[s]$
Δx^{safe}	safe distance headway $[m]$
v^{safe}	safe velocity in $[m/s]$
τ	driver reaction time in $[s]$
μ	stochastic parameter in $[0;1]$

Table 2.1: Vehicular Mobility Model Description

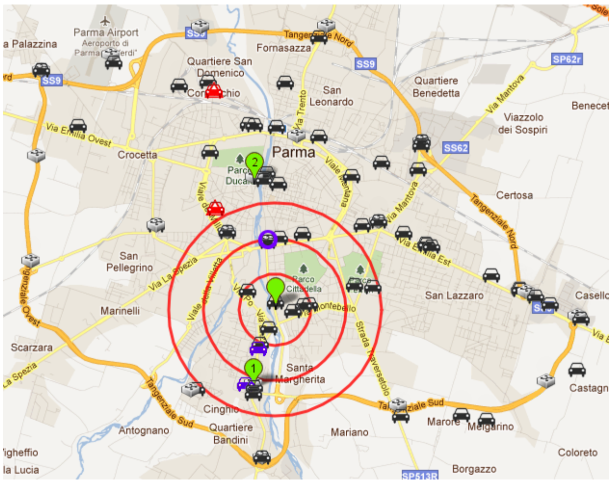


Figure 2.11: Example of simulated DGT-based VSN (in the city of Parma).

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 [41], Krauss [30], and IDM [44]). 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:

$$v^{\mathrm{des}} = \max\left\{v_{\min}, v^{p}_{\max}\left(1 - \frac{k}{k_{\mathrm{jam}}}\right)\right\} \tag{2.11}$$

where $v_{\min}$ is the minimum car speed (depending on vehicle characteristics), $v^{p}_{\max}$ 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 it 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:

$$\begin{aligned} v^{\mathrm{safe}} &= \tfrac{d^2}{k_1} + k_2 \\ k_1 &= \tfrac{d^2_{\mathrm{limit}}}{v^{\mathrm{des}} - v_{\min}} \\ k_2 &= v_{\min} \end{aligned} \tag{2.12}$$

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.

2.7.3 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 figure 2.12). To this purpose, we use one model to describe the mobility of vehicles, and another model for taking into account vertical handover.

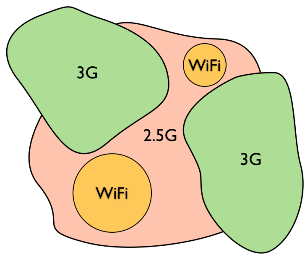
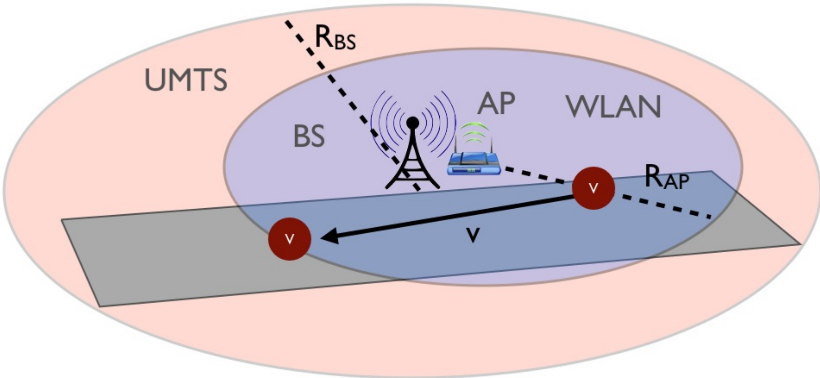


Figure 2.12: Example of area with different, partially overlapping network coverages.

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 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 2.13 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).



(a)

Figure 2.13: Vertical Handover scenario considering two overlapping regions (Cellular Base Station and WiFi Access Point).

Regarding vertical handover, different algorithms for reducing the delay and the packet loss rate have been proposed. The Always Best Connected (ABC) concept has been introduced in [53] and [56] 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 [54] and [55].

The vertical handover algorithm we adopt in our analysis is based on the approach presented by Esposito *et al.* [57], 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 *serving 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:

$$B_{CN} > \frac{B_{SN}}{1 - \frac{L}{\Delta T}} + \delta \tag{2.13}$$

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:

$$\Delta T = \frac{\Delta x}{|v^{des}|} = \frac{R - d(V, CN)}{|v^{des}|} \tag{2.14}$$

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 [58]) .

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.

2.7.4 Simulation-based Analysis

Evaluating the performance of our DGT-based protocol in significant, dynamic scenarios cannot be done analytically, because of the high complexity (non-linearity) of the problem. For this reason we used the discrete event simulation tool called DEUS [63], that provides a simple Java API for the implementation of nodes, events and scheduling processes, and a straightforward but powerful visual tool for configuring simulations of complex systems.

In order the perform realistic tests we set up an integration between DEUS and Google Maps API. With the features provided by Google Maps API we have created a simple HTML/Javascript control page that allows to monitor any simulated node, following it from starting to final position, and all the neighbors in its geo-buckets. This solution allows to study the protocol not only with specific P2P metrics - like message rate, miss ratio, number of peers, etc. - but also with a direct monitoring of peer behaviors during the simulation.

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 20Km) and the other surrounding Parma (square area with side of 7Km). 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 $5Km/h$ and $100Km/h$ 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.

Performance metrics

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/sec]:* Message Rate, i.e. the average number of messages received per second by each node.

- *NPE [Km]:* Node Position Error, i.e. the average distance between a peer's position reference in a GeoBucket and its actual position.

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 1000 peers for a virtual time of 10 hours (corresponding to 10000 virtual time units) while the second one has doubled size (2000 peers) and time span (20 hours, that is 20000 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.

The following table presents all the considered cases.

Case	#geo-buckets	GB thickness	#Peer	final VT
1	10	1.5 Km	1000	10000
2	5	3 Km	1000	10000
3	10	0.5 Km	1000	10000
4	10	1.5 Km	2000	20000
5	5	1.5 Km	2000	20000

Fig.2.14 shows the PMN for cases 1, 2 and 3, which refer to the first peer system (1000 nodes over 10 hours). 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

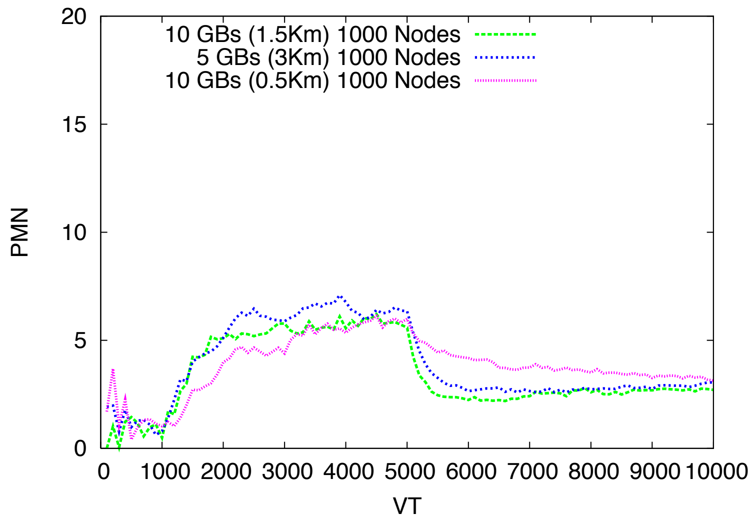


Figure 2.14: PMN - GB configuration evaluation (cases 1, 2 and 3) with 1000 nodes.

to around 5%, while it decreases significantly when the peer network reaches a more stable state.

Another important metric that we must take in account to evaluate these different configurations is the MR. Fig.2.15 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 (2000 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.

Results in Fig.2.16 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 1.5Km thickness are used, which means a covered area

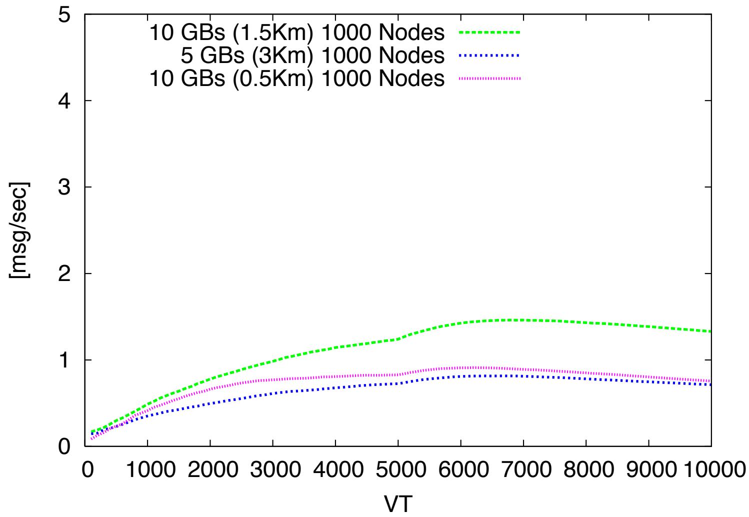


Figure 2.15: Global MR - GB configuration evaluation.

of 706Km^2, whereas in case 5 we have only 5 geo-buckets with the same thickness for a covered area of 176Km^2. There an evident difference in the covered area but the performance is very good in both cases. The little amount of missing nodes depends on the dynamics created by new incoming peers and by the high rate of movements generated by nodes traveling on their paths.

In order to provide this level of performance with different configurations and covered areas, the protocol needs to route messages to users in the target zone. The denser scenario, as described for the smaller network, implies a different amount of exchanged messages (Fig.2.15) that depends on peer density in the analyzed area.

We observe that the accuracy of the protocol shows little dependance 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

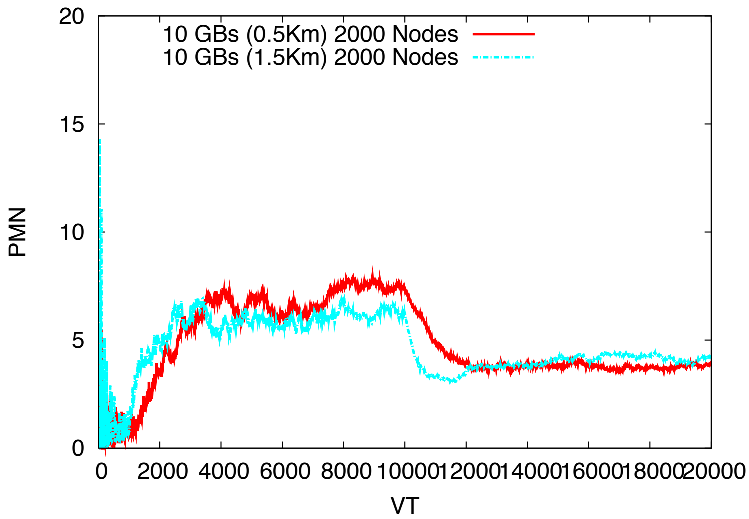


Figure 2.16: PMN - GB configuration evaluation (cases 4 and 5) with 2000 nodes.

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.2.17 is related to case 1, with 10 geo-buckets having a 1.5Km thickness. We already showed the associated PMN in Fig.2.14 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 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 applications like Vehicular Networks, where it is very important to have the best knowledge of active users in a

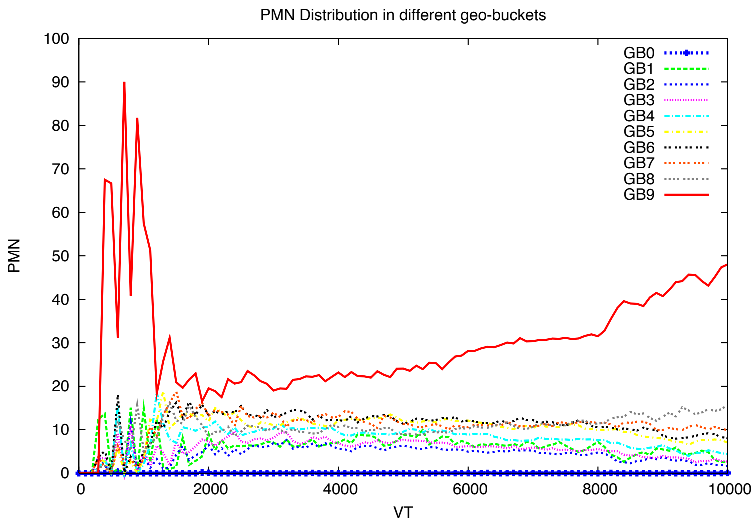


Figure 2.17: PMN Distribution in each geo-bucket - GB configuration evaluation (case 1).

specific area of interest around the car.

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 $\approx$ 5400 peers over a virtual time of 50 hours using 10 geo-buckets with 1.5 Km thickness.

Fig.2.18 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.2.19) 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

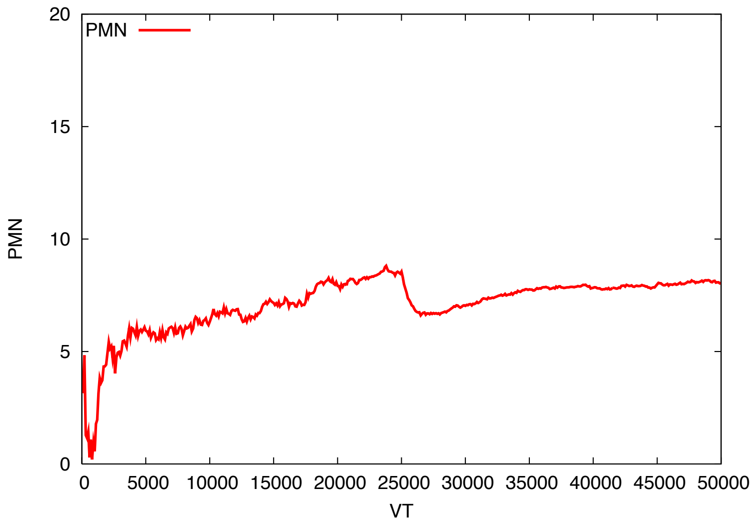


Figure 2.18: PMN - Larger network.

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). 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 GB * 1.5Km) - 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.

Evaluation of the Position Update mechanism

This scenario was created to assess the effects of ε variations on protocol performance. Using a network of 2000 nodes (10 GBs and a thickness of 1.5Km), we ran multiple simulations by varying ε between 0.1 Km to 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

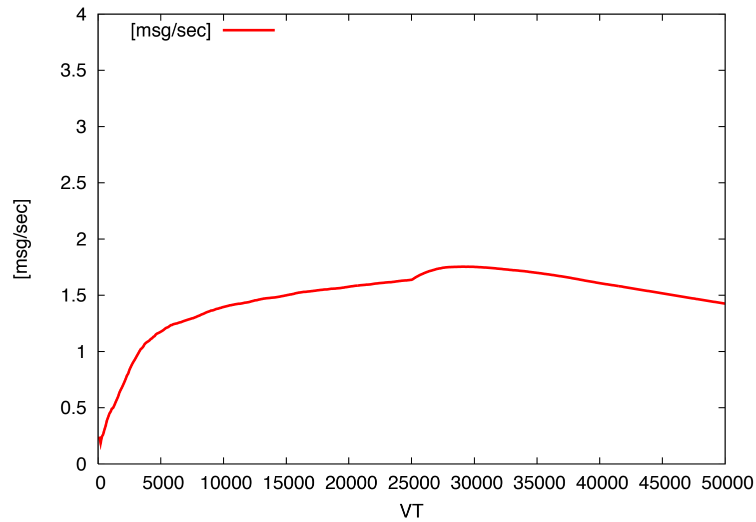


Figure 2.19: MR - Larger network.

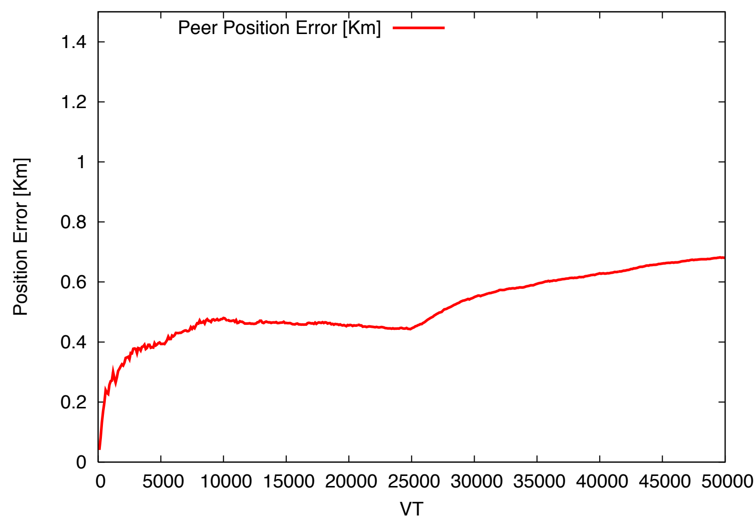


Figure 2.20: NPE - Larger network.

when users change their locations, whereas a high value causes infrequent updates.

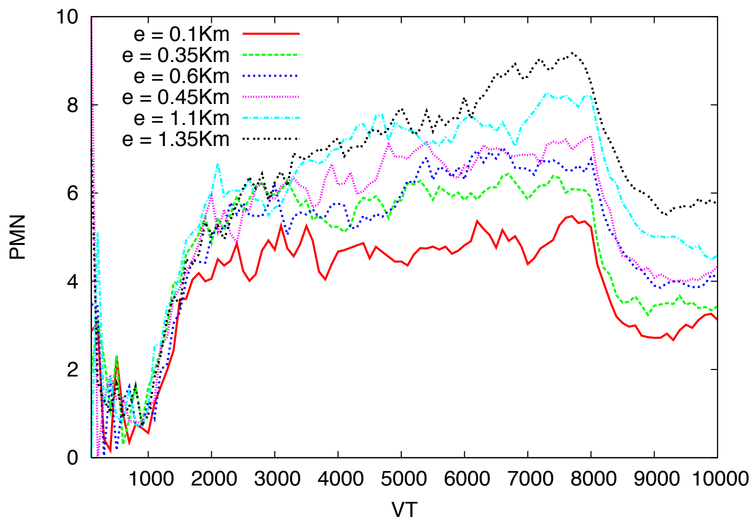


Figure 2.21: PMN - Position Update Evaluation

The accuracy of information stored in GBs is clearly related to the value of ε. Fig.2.21 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.2.22) 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.

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. Fig.2.23 shows the results of different configurations and suggests that a value between 0.35 Km and 0.6Km 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

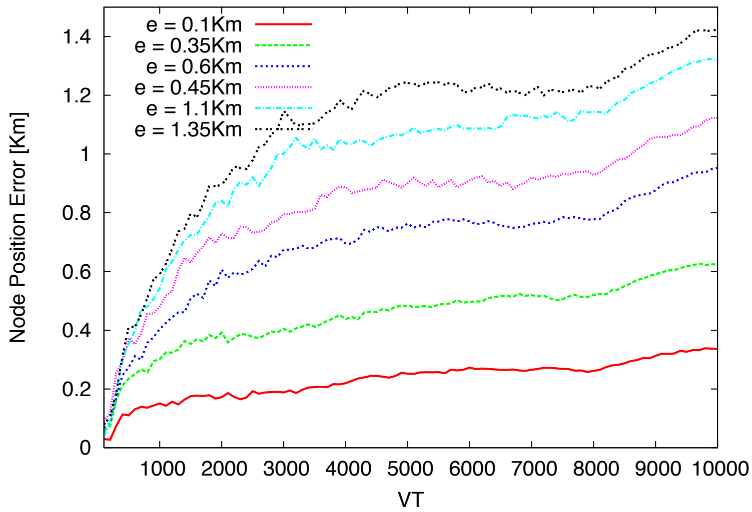


Figure 2.22: NPE - Position Update Evaluation

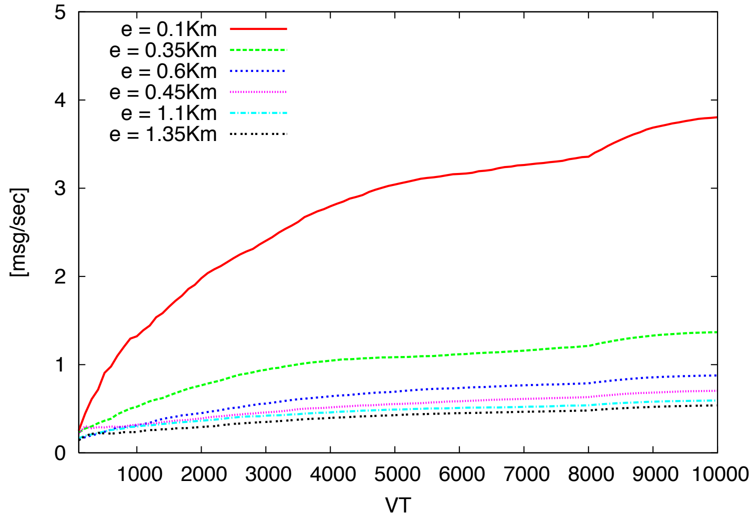


Figure 2.23: MR - Position Update Evaluation

opportunity to reduce the number of exchanged messages without a great impact on global accuracy.

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 vehicular network mobility model using generated paths around Parma. The network is characterized by 1000 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 $\varepsilon = 0.1$. A varying degree of node disconnection and the related PMN distribution have been analyzed. Fig. 2.24 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. 2.25 that illustrates how the PMN is distributed in GB_0 revealing that is always very low and not significantly affected by peer disconnections.

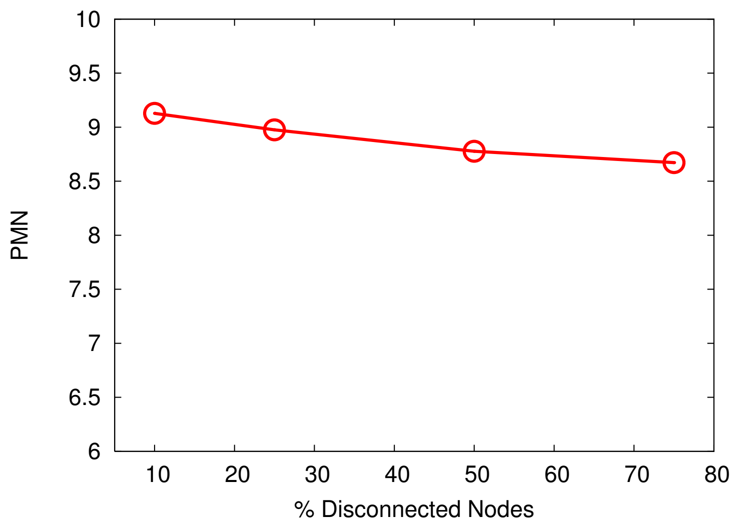


Figure 2.24: *PMN* for different percentages of disconnecting nodes.

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 modeled mobility scenario. The second part aims at evaluating how the size of the peer's local region of interest (as expressed by the number of 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 0.5Km, $\varepsilon = 0.1$ as well as a dynamic discovery period with a range of $[1.5;6]min$ as in previous analysis.

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

Simulation life is initially characterized by a growing number of active users that step by step join the network, start moving, exchanging messages and discovering

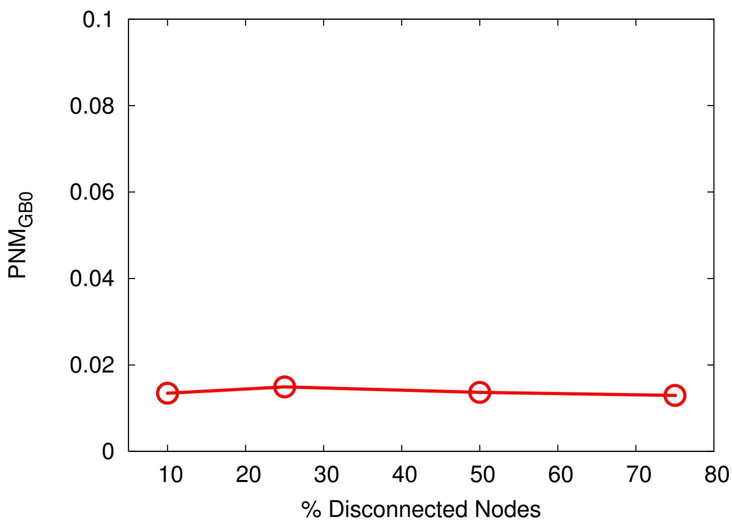


Figure 2.25: PMN_{GB0} for different percentages of disconnecting nodes.

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. Fig.2.26 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 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. 2.27 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 $\cong 28Km^2$) 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. Fig. 2.28 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.

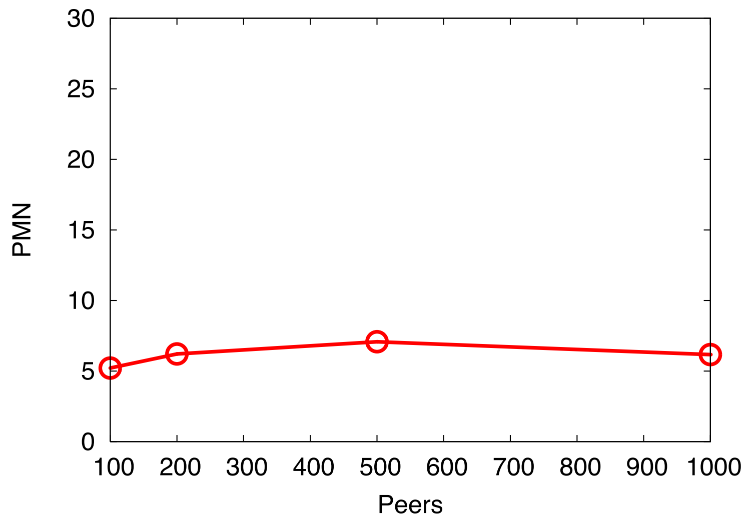


Figure 2.26: *PMN* value for different network sizes.

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 10km^2-squared area centered on the city of Parma and the mobility model previous 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 min 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 (10000 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 $\delta = 1[Mbps]$ as hysteresis value to avoid handover if the two competing networks have negligible data rate difference. Road accident events are scheduled

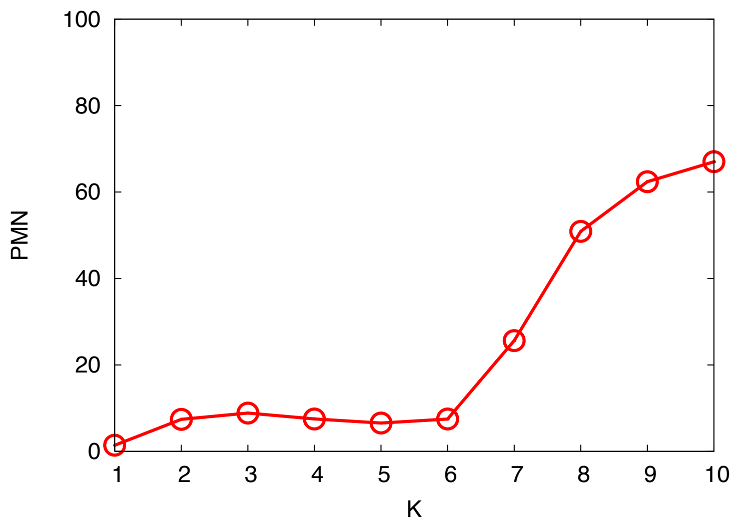


Figure 2.27: *PMN* results for different K values.

during the simulation according to a Poisson process with mean inter-arrival value of 1000 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 $x\%$.

- $PacketLoss/min$: 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.

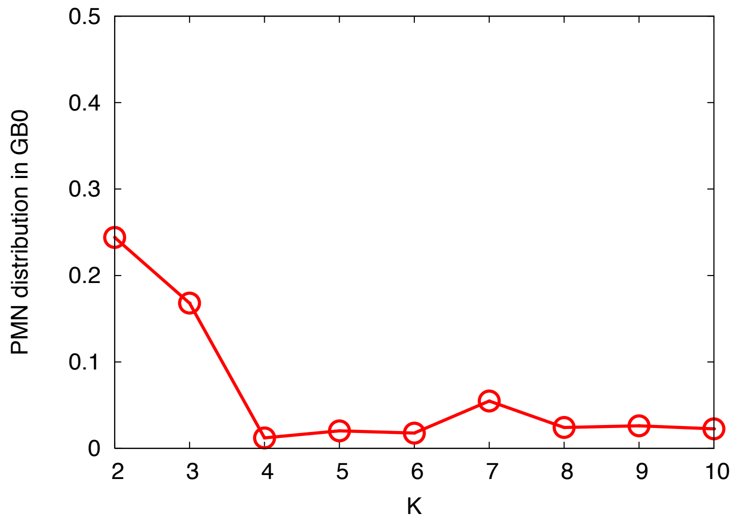


Figure 2.28: *PMN* distribution in GB0 for different K values

With the aim of improving the accuracy and the realism of simulated models, we performed multiple field measurements using Android smart-phones (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 used to build a map of available cell towers (the overall coverage is schematically shown in figure 2.29), each one characterized by a specific connectivity type and a coverage area with a 1.5Km radius. A WiFi region with a radius of 2.0Km is also

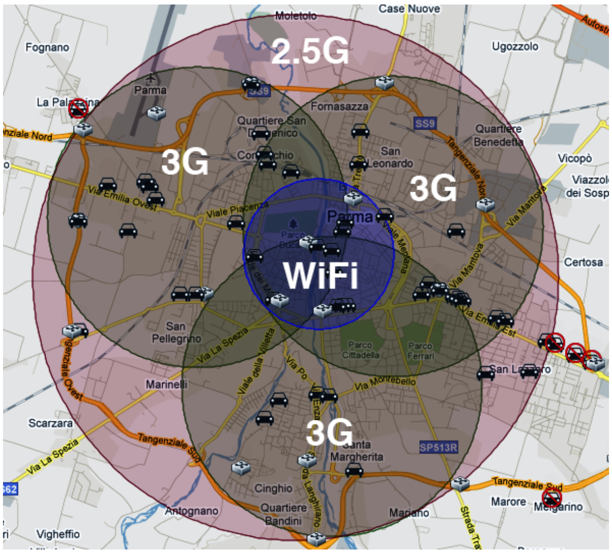


Figure 2.29: Simulated network regions in Parma urban area.

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 2.2 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 net-

#	Type	Uplink min-max [Kbit/s]	Downlink min;max [Kbit/s]
1	2.5G	30-90	60-170
2	3G	35-1150	91-2650
3	WiFI	100-2000	2000-10000

Table 2.2: Types and performance of the available network regions in the simulated urban area.

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

Table 2.3: Vertical handover latency timetable.

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%

Table 2.4: Simulated scenarios with different connectivity coverage of the urban area.

work coverage, considering the vertical handover latencies reported in Table 2.3. Simulated coverage distributions are summarized in Table 2.4, 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 figure 2.30 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

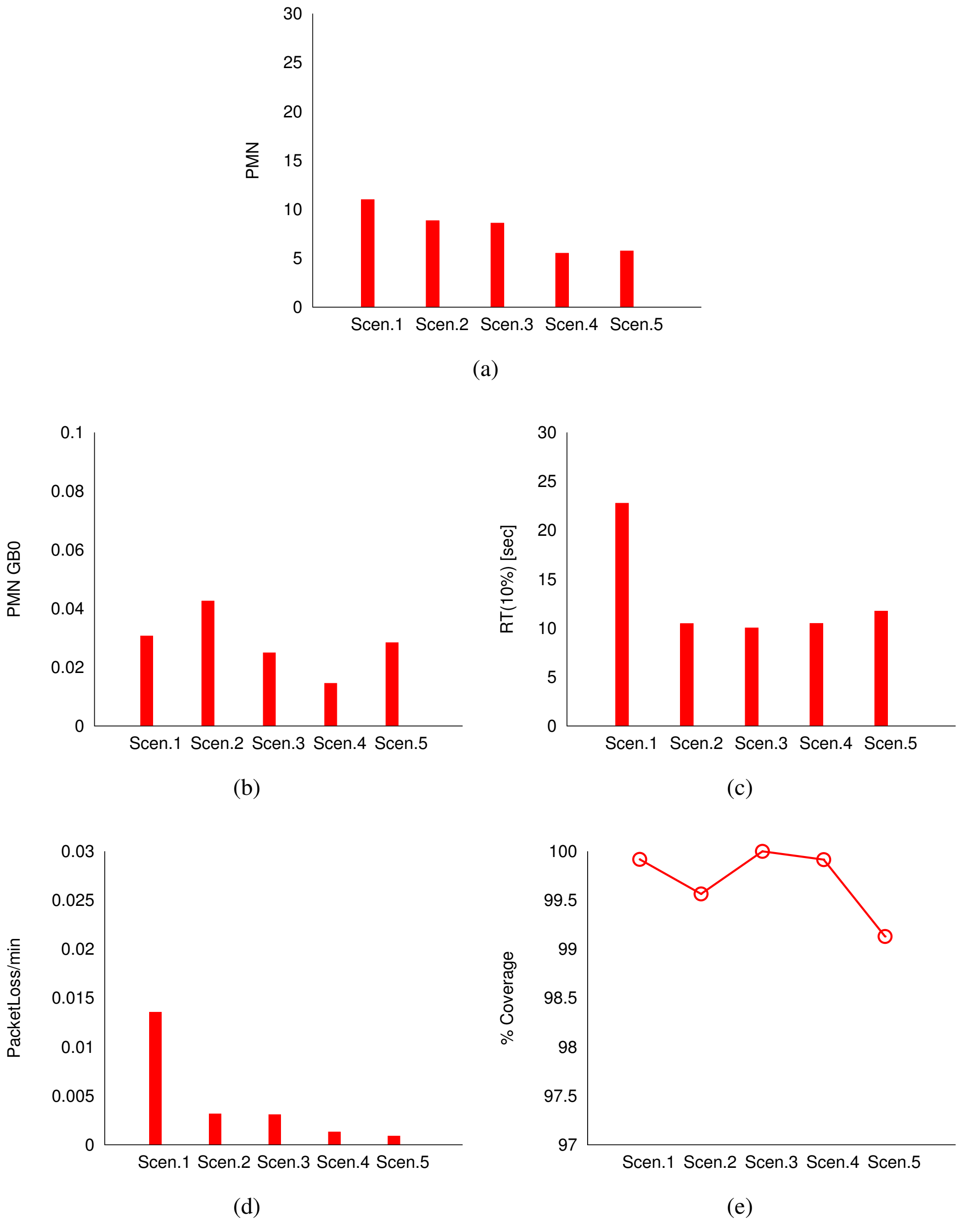


Figure 2.30: Results related to different simulated scenarios. (a) PMN, (b) PMN in GB0, (c) RT(10%), (d) PacketLoss / min, (e) % Coverage of traffic information messages

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, *e.g.* [57]), 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 figure 2.31(c)), and consequently the global percentage of missing nodes (figure 2.31(a)), that proportionally grows with the latency. The same behavior can be observed for the number of lost packets per minute (figure 2.31(d)) that however remains globally small. As presented in the analysis related 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 figure 2.31(b) and (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%.

2.8 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),

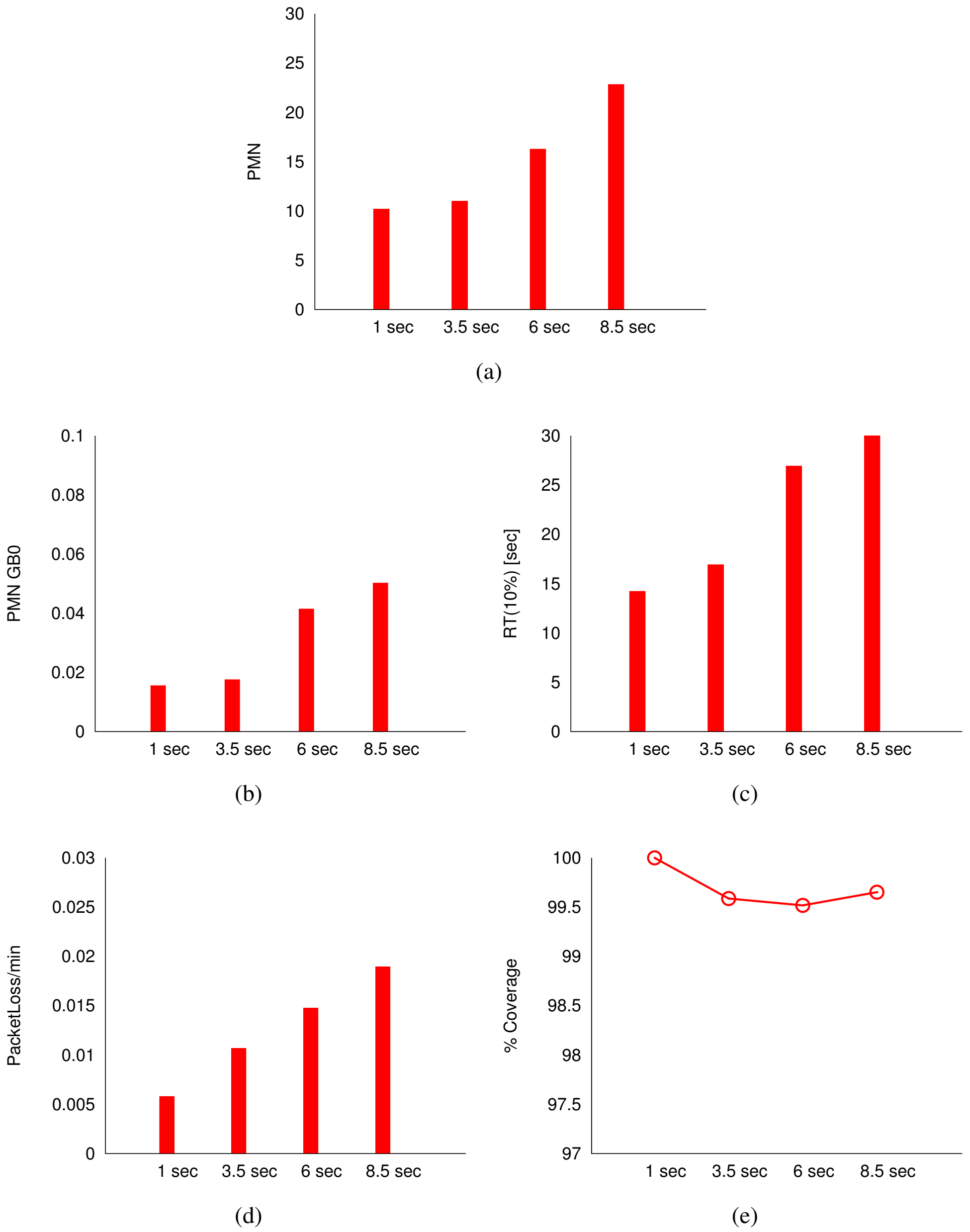


Figure 2.31: Results related to different latency values. (a) PMN, (b) PMN in GB0, (c) RT(10%), (d) PacketLoss / min, (e) % Coverage of traffic information messages

- 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 an inter-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.

Chapter 3

Applications

In this chapter we present the application of the DGT approach to two different scenarios related to the city environment. Firstly, we used the designed overlay to build a distributed traffic information system (TIS) for the dissemination of traffic alert messages. In such a system 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. The second application is a joint DGT and network coding approach for large scale information management in smart cities. In this scenario we envisioned that user can generate and consume relevant information about their statuses to enhance the security and lifestyle of their citizens. In this context, our challenge is how the information can be maintained and distributed efficiently among the city itself.

3.1 Traffic Information System and Vehicular Network

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 [26] has a prominent role, paving the way to several valuable applications, namely geocasting, mobile data

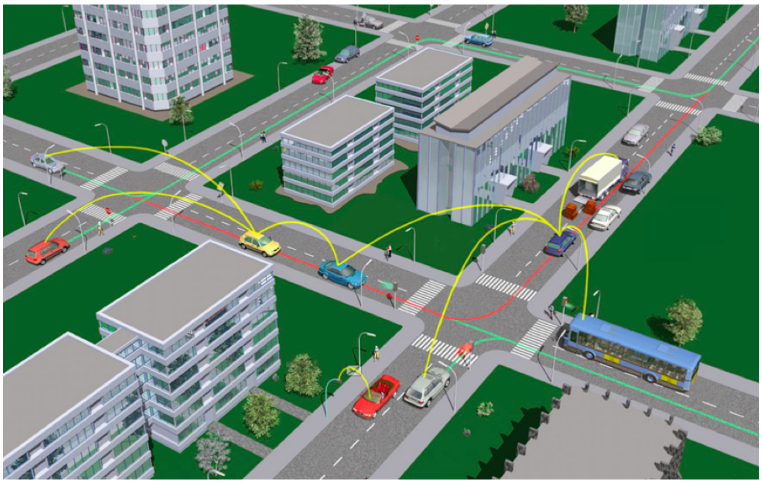


Figure 3.1: V2V and V2I scenarios

sensing and storage, street-level traffic flow estimation, etc. [32]. Vehicular networks builds upon Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) connection, as well as on hybrid variants [26].

Vehicular Sensor Networks (VSNs), that 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 sensing platforms [21]. 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 standard [52] 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 *peer-to-peer* (P2P) approach is highly appealing. Initially exploited within V2V schemes [34], P2P approaches have been recently followed also for realizing decentralized traffic information system [40]. 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 [39].

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.

3.1.1 VANET Background

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 [32]. An example is MobEyes [31] that proposes a strategy for harvesting, aggregating, and distributing sensed data by means of proactive urban 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) [29], WiMax/802.16e [24], WLAN [23], as well as cellular systems [38]. 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. In [27], Hull *et al.* point 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 it own measurements regularly).

For these reasons, in recent years researcher 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 [39] and, more recently, with PeerTIS [28], 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. [40] shows another P2P approach based on *cellular networks* (CNs) and on the JXTA middleware [51], 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 [36][16][37], allowing applications such as ride quality monitoring, street-level traffic flow estimation, and proactive urban surveillance.

3.1.2 DGT for Vehicular Networks: the D4V Architecture

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 [17] 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 work, 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 [43][15], request/reply [48][46], broadcasting, *sharing* [42] and beaconing [47][20]. 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 [33], 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.

3.1.3 D4V Performance Evaluation

Here, 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: $[\text{Kbyte}/\text{peer} \cdot \text{sec}]$): 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.

The proposed model of VSN has been studied by means of a general-purpose discrete event simulation environment, called DEUS, released by the University of Parma as open source under the GPL license [5].

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&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 [62]). The simulated D4V covers ten hours of system life (10000 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 [73]. 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 12.5Km^2 and a dynamic discovery period ranging from 1.5 min to 6 min depending on the number of discovered nodes.

The first step of the evaluation was oriented to analyze the effect of varying the ε threshold ($\varepsilon \in [0.1; 1.0]$ Km with a step of 0.3 Km) considering two different peer densities $\delta = 10$ peer/Km and $\delta = 20$ peer/Km and a range of interest, for the disseminated messages, of 4 Km. As defined earlier, ε represents the minimum displacement threshold considered by a peer to notify its geographic position update to nodes in the 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.3.2 show simulation results for the considered main metrics, in particular Fig. 3.2(a) 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. 3.2(b) the percentage of cars 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. 3.2(c)) 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 cars 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. 3.2(d) 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 Km to 10 Km to understand how it influences the dissemination process and its cost. Fig. 3.3(a) and Fig. 3.3(b) illustrate the coverage percentage and the total number of cars 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 car 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. 3.3(c), 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

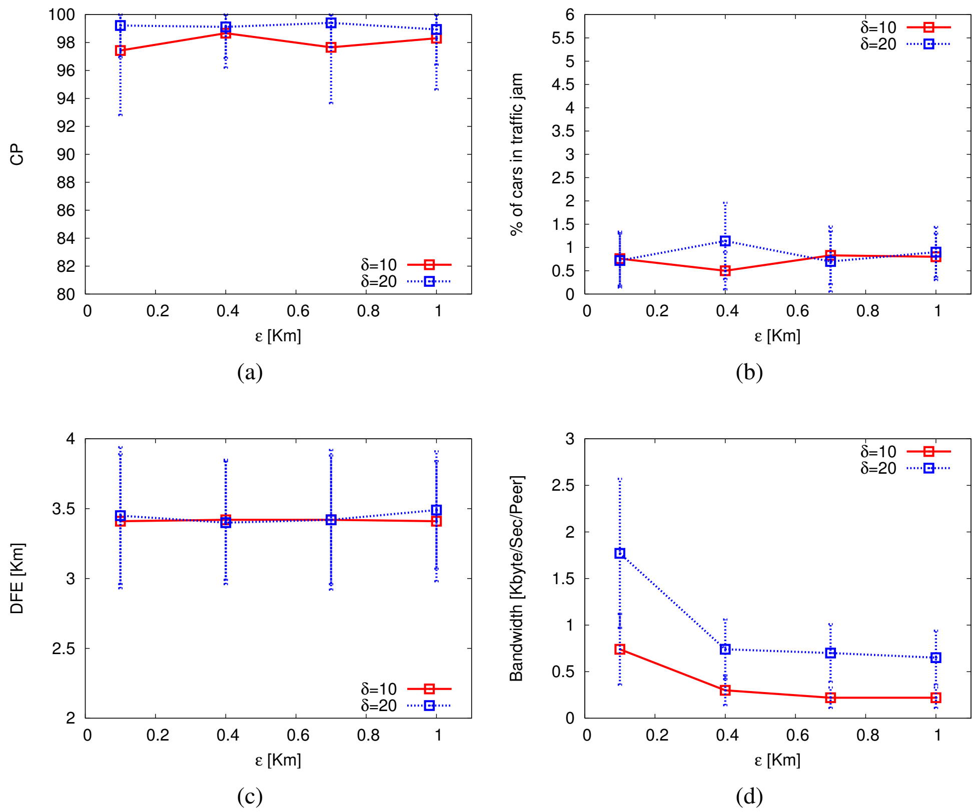


Figure 3.2: Simulation results for different ε values: Coverage %(a), Number of cars in traffic jam (b), Distance From Event [Km] (c), Bandwidth [Kbyte/peer/sec] (d).

quite 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. 3.3(d) — 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 cars, followed by a stable phase without new joins or disconnections. The results in Fig. 3.4(a) confirm that the proposed solution is able to cope with different node densities with no performance degradation, keeping the Coverage Percentage value significantly high (around $98 \div 100$ %) — even in the case of very low density (5 peers/km) which could be quite critical for VANET-based applications. We remind 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. 3.4(c) 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. 3.4(b), the percentages of cars 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 cars that would otherwise grow significantly for increasing density. Fig. 3.4(d) shows the average data traffic per peer (in Kb/sec/peer), as a function of the density, needed to maintain the DGT overlay and disseminate traffic information messages to other active neighbors. Even if there is a moderate and natural increase associated with the growth of nodes, the amount of

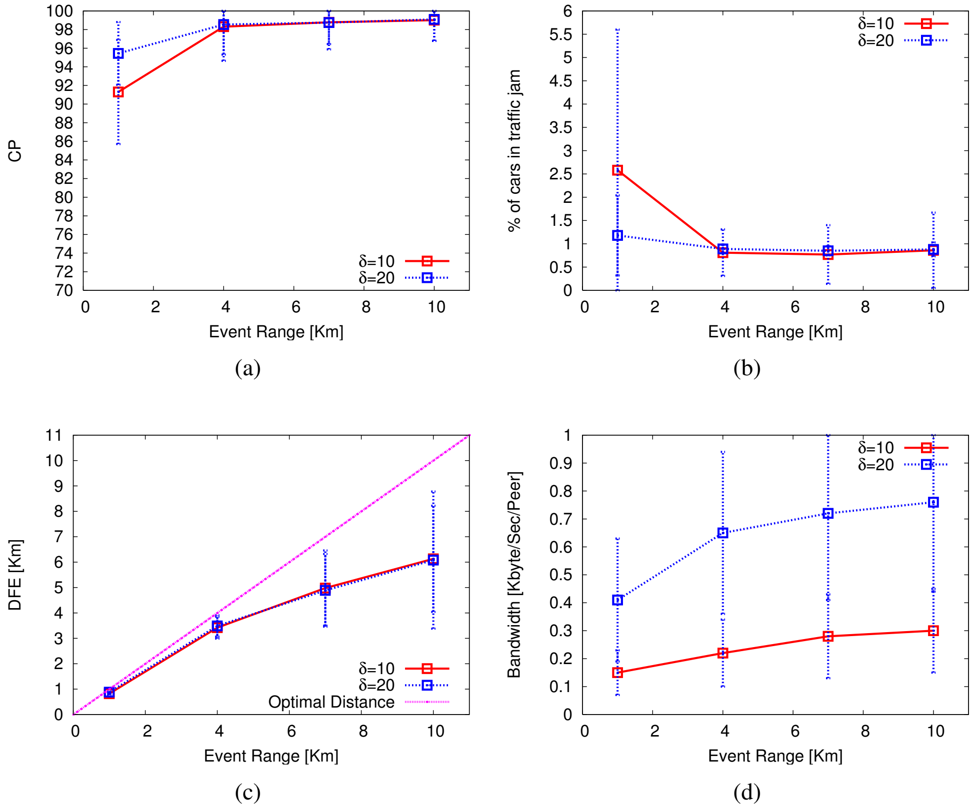


Figure 3.3: Results for different values of the dissemination range: Coverage % (a) , Number of cars in traffic jam (b), Distance From Event [Km] (c), Bandwidth [Kbyte/peer/sec] (d).

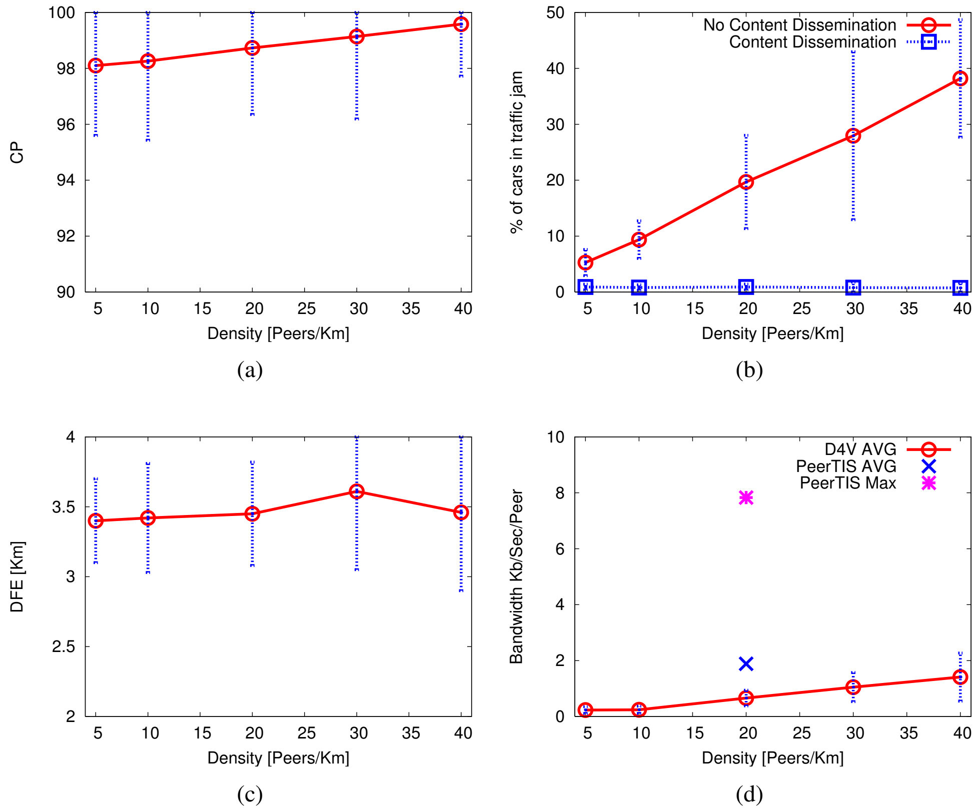


Figure 3.4: Simulation results for different peer densities: Coverage %(a), Number of cars in traffic jam (b), Distance From Event [Km] (c), Bandwidth [Kbyte/peer/sec] (d).

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. For comparison purposes, in Fig. 3.4(d) the performance of PeerTIS [28], is shown: it can be concluded that our technique, for the same peer density conditions, performs significantly better. Furthermore, for all analyzed scenarios, the required mobile connection rate easily fits the theoretical limits of UMTS networks (384 kbps for upload) and older cellular systems, like EDGE (236.8 kbps for four time slots) or GSM/GPRS (up to 114 kbps).

Fig. 3.5 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. 3.5(a) 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. 3.5(b), the percentage of cars 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 car really low. D4V robustness is also confirmed by the results in Fig. 3.5(c), 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. Fig. 3.5(d) reports the data consumption, which is unavoidably lower than in the other scenarios due to the lack of a recovery procedure for lost packets.

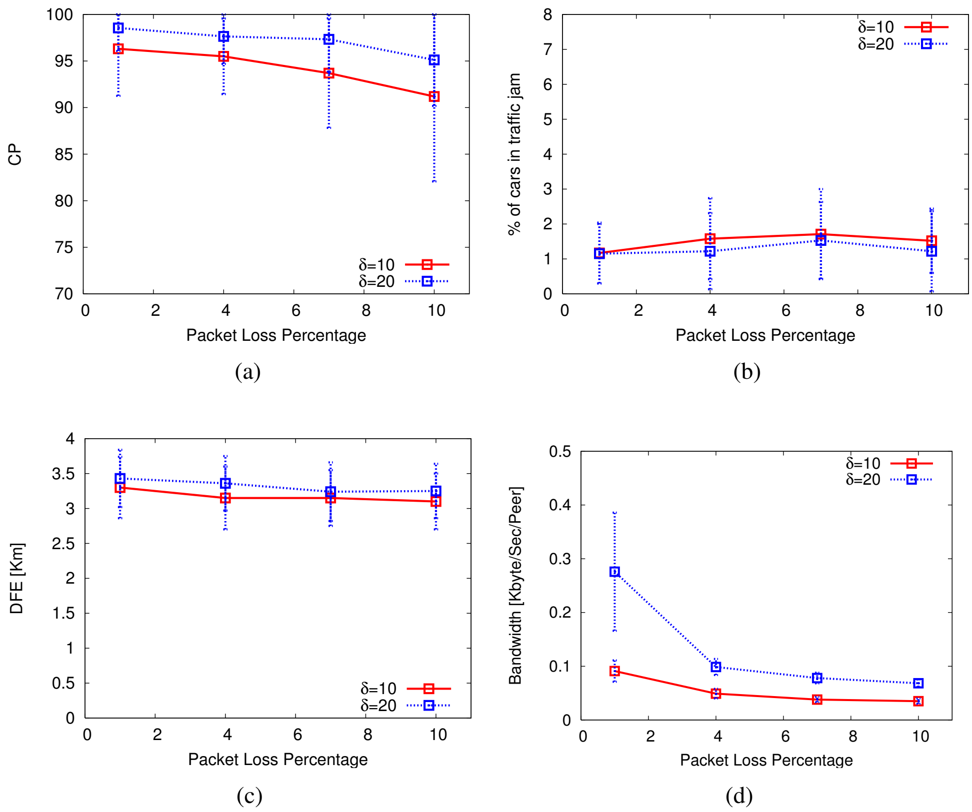


Figure 3.5: Results for different Packet Loss Percentage: Coverage % (a) , Number of cars in traffic jam (b), Distance From Event [Km] (c), Bandwidth [Kbyte/peer/sec] (d).

Finally, considering the behavioral model of a driver in proximity of a road stretch with a bad surface condition, that we have introduced in Section 2.7.2, in Fig. 3.6 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.

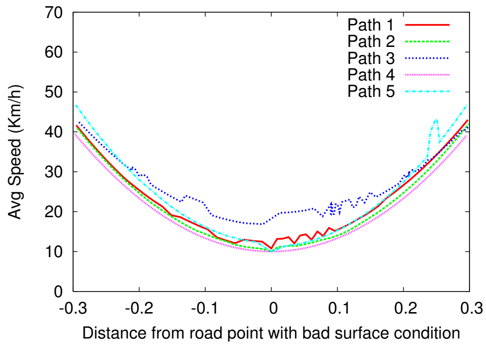


Figure 3.6: Average of driver speed near road points with bad surface condition.

3.1.4 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

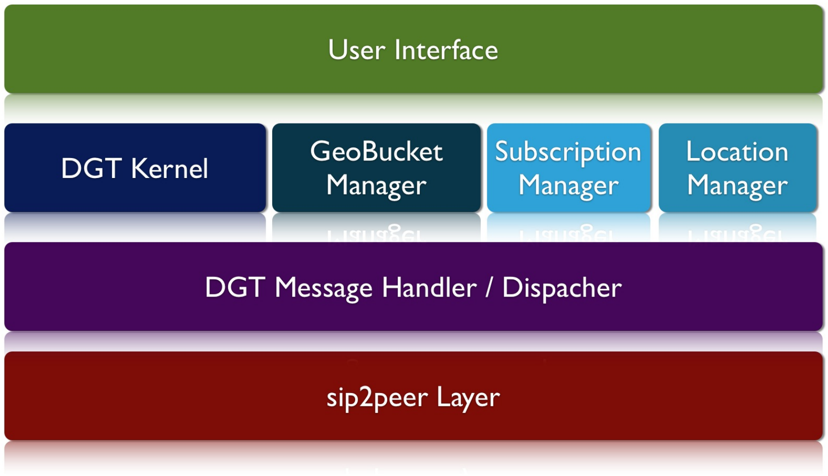


Figure 3.7: D4V Prototype modules

approaching dangerous situations. The development of the DGT library started from our novel peer-to-peer middleware called *sip2peer* [77], that 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* [79] 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 3.7 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/Dispacher (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 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.3.8(a)). 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.3.8(b)). 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.3.8(c)). 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 meters, and notify with a dialog message the alert type and its location (Fig.3.8(d)(e)). Furthermore, the user can review in any moment the list of received messages (Fig.3.8(f)) sorted by distance, visualize them on the map, renew expired content or report an abuse or false information generated by a user.

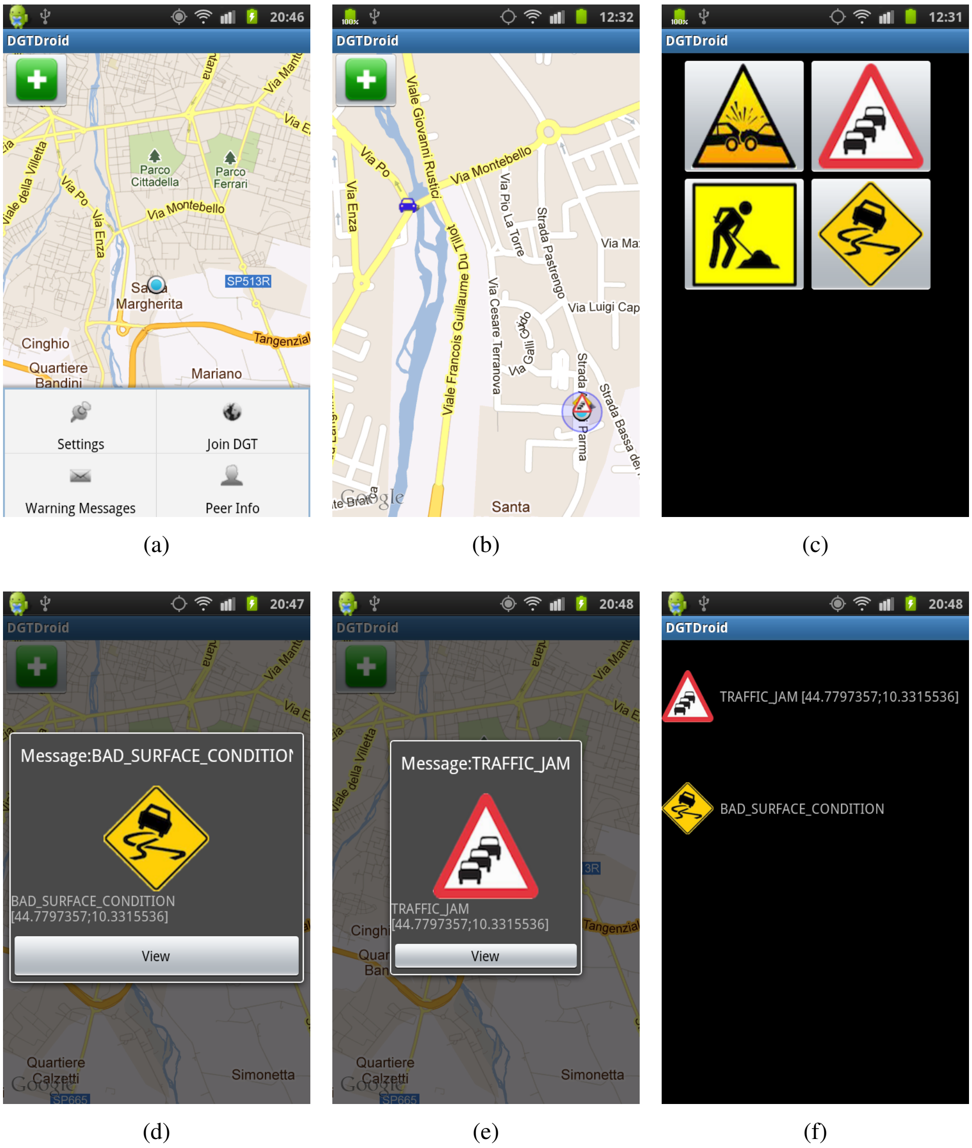


Figure 3.8: Android DGT Prototype (a) Settings Menu. (b) Map View. (c) Traffic Message creation view. (d)(e) Bad Surface Condition and Traffic Jam Messages popup view. (f) Incoming message list.

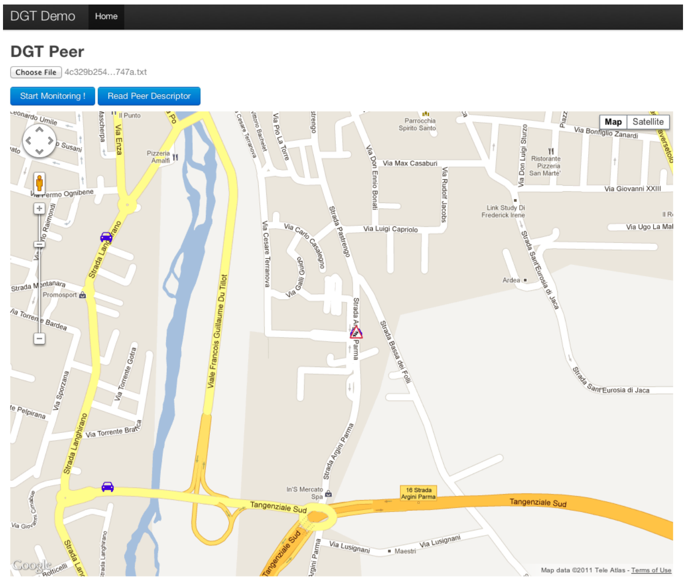


Figure 3.9: D4V Prototype Web Monitoring Tool

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 a 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.3.9) 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.

3.1.5 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 (Figures 3.10(a)(b)). Peers

(a) (b)

Figure 3.10: Car Setup (a) Incoming warning. (b) Traffic Message creation.

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 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, distributed hash tables, and query processing. PlanetLab currently consists of 1089 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 seconds 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 seconds containing all the required statistic for the performance evaluation.

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

- *Bandwidth* (dimension: [Kbyte/peer · sec]): 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 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.

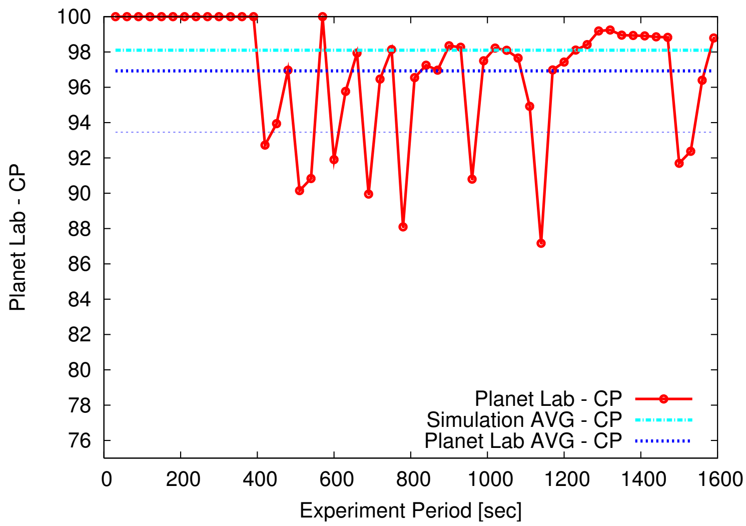


Figure 3.11: PlanetLab - Coverage Percentage

Graph in Figure 3.11 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 (≅98%). The CP curve shows that when new messages are generated the coverage percentage goes slightly down to lower value (≅88%) but after one or two time line steps (30/60 seconds) recovers to an high coverage percentage confirming 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 3.12 illustrates the DFE trend comparing PlanetLab and simulation average, that also in this case are really close and around (≅3.5 Km). The graph confirms that vehicles that did not receive the mes-

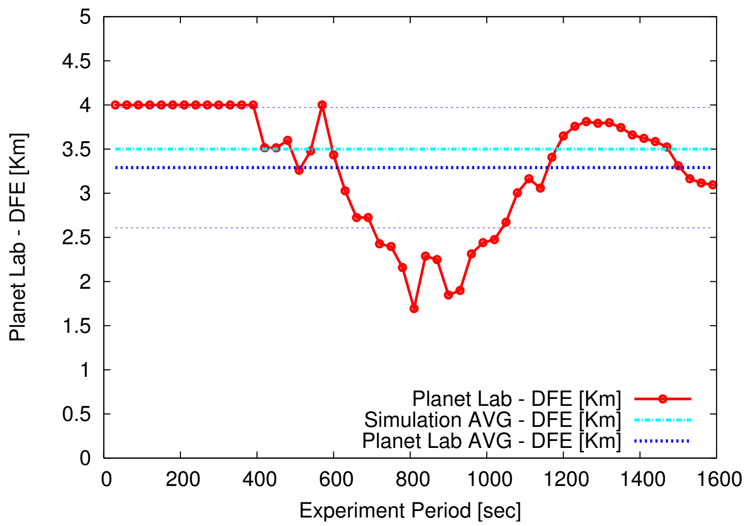


Figure 3.12: PlanetLab - Distance From Event (DFE)

sage 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 Figure 3.13 reports the bandwidth in terms of Kb/Sec/Peer during the experiments conducted on PlanetLab. In this case the average value (≅0.3 Kb/Sec/Peer) is different from the same value obtained during the simulation (≅0.9 Kb/Sec/Peer). This difference is 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. Notwithstanding this lightly distance between such D4V values the PlanetLab result is still lower than the average value of PeerTIS [28] of ≅1.8 Kb/Sec/Peer.

Graphs in Figures 3.14 and 3.15 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

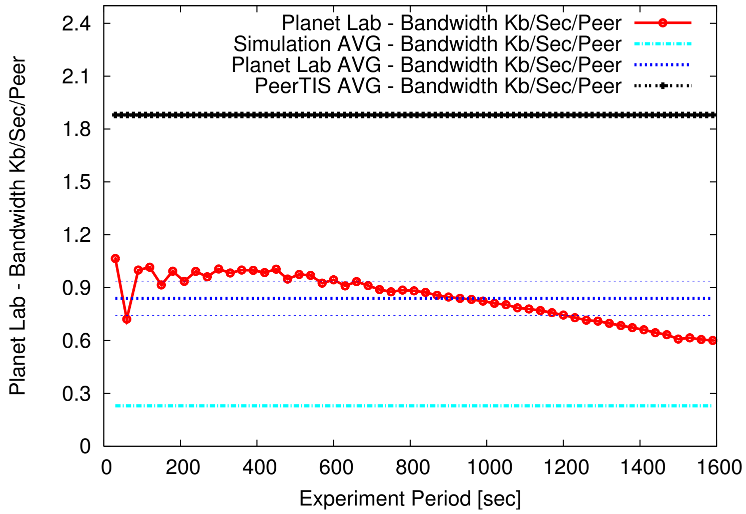


Figure 3.13: PlanetLab - Average Bandwidth

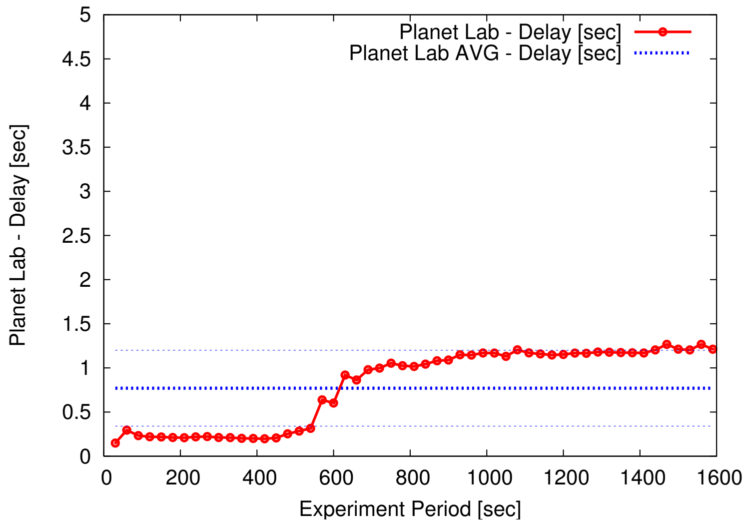


Figure 3.14: PlanetLab - Average Delay

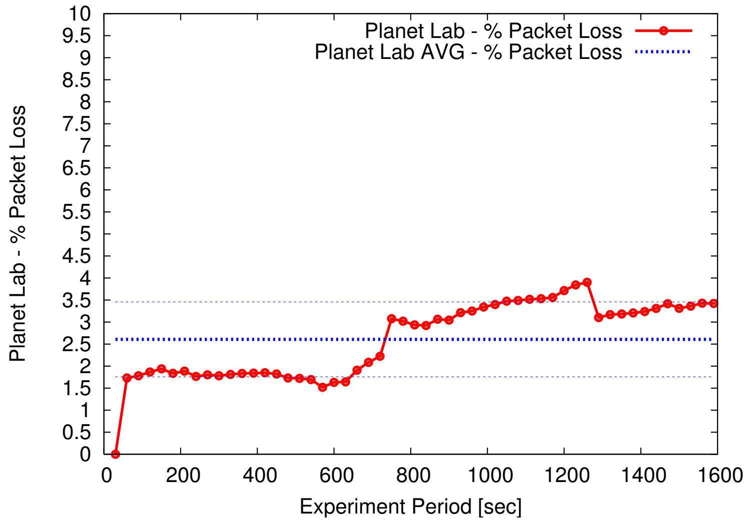


Figure 3.15: PlanetLab - Average Packet Loss

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).

3.1.6 Large Scale Information Management in Smart Cities

In this section we present our architecture for large scale information management in smart cities. We envisioned that users can generate and consume relevant heterogeneous information about their statuses to enhance the security and lifestyle of their citizens. In this context, our challenge is to use the DGT algorithm to maintain, discover and efficiently distribute raw data and aggregated information among the city itself.

According to a study of the United Nations [81], the fraction of the world's population living in cities is going to increase from current 50 to 70 percent, due to growth in the current urban population and migration from rural areas. Indeed, cities offer large economic, social, and political opportunities, as well as potential for significantly greater environmental sustainability. However, it is necessary to find new ways to manage complexity, to increase efficiency, to reduce expenses, and to improve quality of life. In other words, cities need to get smarter.

Progress lies in an accurate view across urban infrastructure, the right level of intelligence to optimize resources, and the ability to integrate information from all departments to anticipate and respond to events. Smarter city transformation relies on the use of powerful analytical techniques to extract insights from real-world events in order to improve urban business processes [71]. Creating and applying a unified information model gives the possibility to obtain a more complete picture of urban activity. The ability to understand how combinations of factors contribute to, e.g., a rapid increase in the demand for water or an unusually high accident rate on a stretch of road in turn facilitates better operational decisions.

The availability of massive amounts of sensed information provides fascinating opportunities to understand city activity by means of modeling and analytics. In this work, we focus on how the information sensed by a smart city can be efficiently maintained and distributed. A large amount of information (either raw or aggregated) generated by city actors (both humans and machines) needs to be stored, maintained, and returned, either in a proactive or reactive manner, to city actors themselves. The approach we propose for maintaining and distributing information consists in defining exchanging information architectures, e.g., overlay networks, which allow to store and distribute city status information with minimum overhead and high reliability. Regeneration is also necessary to improve the robustness of the system against possible data losses.

To this end, we envision the integration of innovative management networks based on the key concepts of peer-to-peer (P2P) and network coding (NC). While the former optimizes the load balancing and avoids the presence of bottlenecks and single points of failure, the use of NC techniques leads to the presence of redundancy, which promises to make any information retrieval extremely reliable and real-time streaming highly efficient. Indeed, NC improves the performance of P2P content sharing systems since it mitigates the block transfer scheduling or piece selection problem, especially when nodes dynamically join/depart from the Internet. Moreover, NC is important also for another functionality of the system, i.e., distributed storage: should a storage node fail, the stored information could be retrieved by properly combining the information contained in other storage nodes.

The proposed architecture includes different types of peers - data sources, storage nodes, data aggregators, user nodes - that may be fixed or mobile. Such peers are organized in a structured overlay scheme called Distributed Geographic Table (DGT), that takes into account their geographic position, thus enabling a number of city-tailored services. A possible application of interest is a traffic information system that supports vehicular mobility, by suggesting alternative routes to avoid traffic jams or accidents. In this scenario, the knowledge of the information geographic position may be useful in computing the routes to be suggested to the users.

3.1.7 Background

- *Smart Cities* The concept of smart city has recently emerged from the interaction of research areas like *intelligent cities* [80] and *smart communities* [61]. Cities can be considered as systems of systems and there are emerging opportunities to introduce digital nervous systems, intelligent responsiveness, and optimization at every level of system integration [76]. For instance, automobiles can participate in the mobility Internet for sharing traffic and travel data. In this context, one major aspect being discussed in the research community is centralization versus decentralization for storing and retrieving data. In particular, robustness and security can be achieved by using a highly decentralized approach [40, 28, 76].

- *Peer-to-Peer* The P2P paradigm enables two or more entities to collaborate spontaneously in a network of equals (peers) by using appropriate information and communication systems without the necessity for central coordination. In the last decade, P2P has been studied and applied to different application scenarios, from file sharing to live multimedia streaming [14]. Recent research is focusing on P2P-based large-scale storage systems [72], distributed hash tables [75], social networks [69], and measurements of real systems [70].

Interisting P2P approaches are those based on geographic localization, e.g, Globase.KOM [67], and those based on traffic information, such as [28].

- *Network Coding* NC is a recently proposed network-oriented channel coding paradigm, arisen in the field of information theory, which generalizes the classical concept of routing in wired networks. With NC, in fact, intermediate nodes are not only allowed to forward incoming packets, but also to encode them. This allows to achieve the multicast capacity [59] and, therefore, leads to potential advantages in terms of bandwidth and computational efficiency, robustness, etc. Although NC has been extensively studied from a theoretical point of view, several practical scenarios, where benefits can be observed, have been proposed in the last years [65, 78]. Examples of practical scenarios of interest are in the field of P2P networks [66] and distributed storage [64]. In our previous work [68], we have illustrated how NC and peer-to-peer can be enabling technologies for robust distributed storage.

3.1.8 Architecture

In this section, we illustrate our distributed architecture for the management of information flows in smart cities. We first present an extend version of the DGT, that allows every node to maintain knowledge about surrounding peers, as well as to publish and retrieve data items. We introduce a new discovery procedure based on the direction of the target and we also illustrate how NC techniques are used to assure data survival.

The routing strategy can be described as a function

$$R : P \times W \times A \rightarrow 2^{p} \tag{3.1}$$

that returns the set of neighbors $N(\omega, a)$ that are in the interest region and satisfy the request. Thus, given a geographic position $\omega \in W$, a node $p \in P$ and a reference region $a \in A$, a routing query has the following structure: $route(p;\omega;a)$. In Fig. 3.16, the routing strategy is used to maintain the neighborhood of a peer, but also to discover active peers in a remote region of interest.

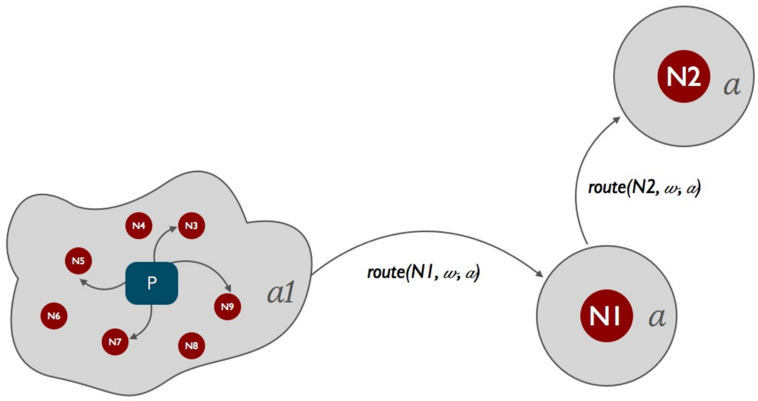


Figure 3.16: Routing of a query.

The query propagation process is affected by the distance between the source and the destination of the query itsely, and by the size of the neighborhood region. The following two types of queries can be designed.

- $N = q_1(lat, lon, type, region)$ returns the list of nodes that are interested on a type of data within a region centered in a specific position, defined by its latitude and longitude. Such a list is then used for publication purposes.

- $A = q_2(lat, lon, type, \Delta t, region)$ returns the list of nodes that own a type of data, in an area that is centered in a specific position defined by its world's coordinate, being such data not older than the specified time range Δt.

Four types of peer are envisioned in the proposed architecture.

- Raw Data Source (RDS), that generates basic data pieces and usually is associated to specific sensors (e.g., traffic cameras, pollution level sensors, etc.); usually, it does not have local storage capabilities.

- Storage Node (SN), which is able to store large amounts of data.

- Aggregated Data Source (ADS), which is a consumer of data produced by RDS, but also producer of filtered/aggregated data.

- User Node (UN), which is interested in receiving and visualize raw or aggregated data in real time, related to any point of interest.

Each peer has a PeerDescriptor that contains the following information:

- node type identifier (i.e., RDS, ADS, UN, SN);
- communication reference (i.e., source/destination IP, port, proxy);
- prioritized list of locations of interest and data types that the nodes want to receive proactively;
- list of generated data types;

Each information element produced by a peer has a DataDescriptor that contains a key, the geographical coordinates of the location in which it has been generated, the data type, and the time validity of the data item. If the data item is a fragment of a larger data item, the DataDescriptor contains also the ordering number. Each information item is finite, has a precise position in space and time, and is univocally identifiable.

The process for publishing a data item includes the following phases.

[Publication phase 1]
If the publisher does not know enough storage nodes, or those that it knows are not available, publication request messages are propagated in a circular region of radius $R_{k_{\max}}$ (see Fig. 3.17). Each message is propagated avoiding nodes that have already received it. Moreover, the number of propagation hops for each request message is reasonably limited.

[Publication phase 2]
The publisher obtains a list of storage nodes that are interested in maintaining the data item.

[Publication phase 3]
From each data item to be published, a set of fragments is generated, using the randomized NC (RNC) illustrated strategy. Such fragments (identified by unique keys

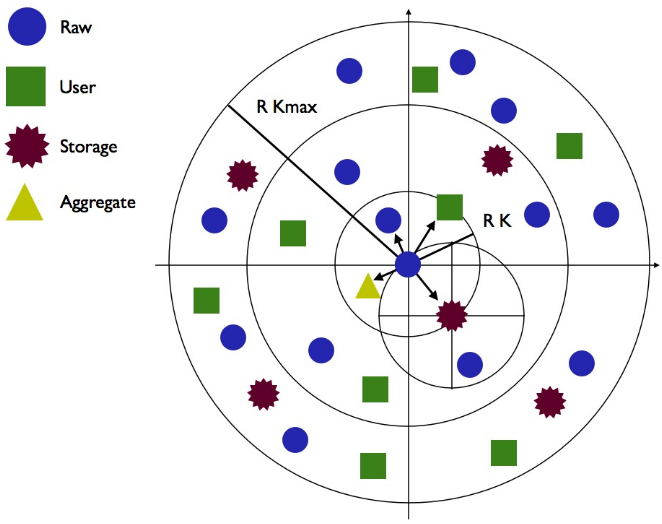


Figure 3.17: Propagation of publication request messages.

and by the key of the data item they have been generated from) are uniformly distributed among peers that belong to the list obtained in the previous phase. In particular, the following algorithm is applied. The area where the fragments have to be published is divided as shown in Fig. 3.18. A fraction of the fragments is published in the small circular region with radius R_k. The other fragments are published in the circular crown whose radiuses are R_k and $R_{k_{\max}}$. Both regions are divided into four quadrants. The fraction of fragments for each quadrant is proportional to the storage node density of the quadrant itself. If a quadrant does not contain storage nodes, fragments are not uploaded there.

[Publication phase 4]

If a quadrant contains more than one storage node, those with more available space are chosen as targets for the publication of fragments.

[Publication phase 5]

When all target storage nodes have been chosen, the peer uploads the fragments in a parallel way.

This process can be generalized. Instead of looking in its neighborhood, the pub-

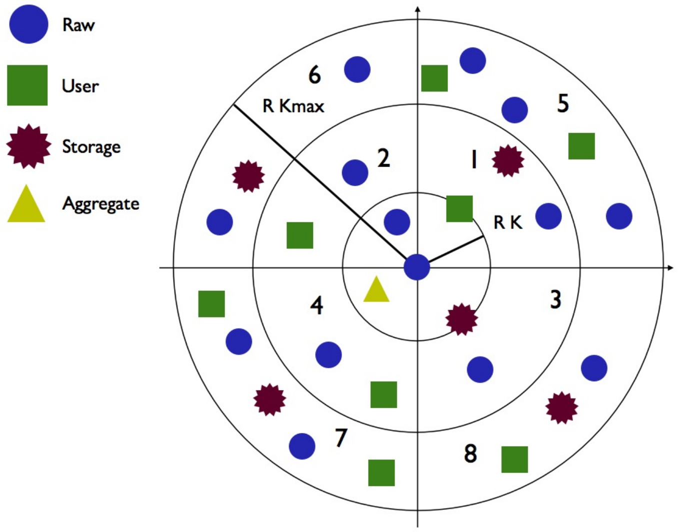


Figure 3.18: Division of the publication area.

lisher may search for storage nodes around any remote location in the known map. Thus, it is possible to implement a publication strategy that takes into account the content that is published. I.e., if the content refers to location *L*, it will be stored nearby *L*, independently on the location of the publisher. Remote publication may be time-consuming, depending on the density of nodes between the publisher's location and the target location. Low density may lead to long and twisted query propagation processes. The search process, described below, is constrained to a limited region between the location of the searcher and the location of interest *L*.

Search Process

Search is illustrated in Fig. 3.19 and is characterized by the following phases.

[Search phase 1]

The node generates request messages for a data item described by a DataDescriptor. Such messages are sent, in the direction of the region of interest, to all nodes within a conic region with dynamic angle α. Query messages return lists of nodes that are

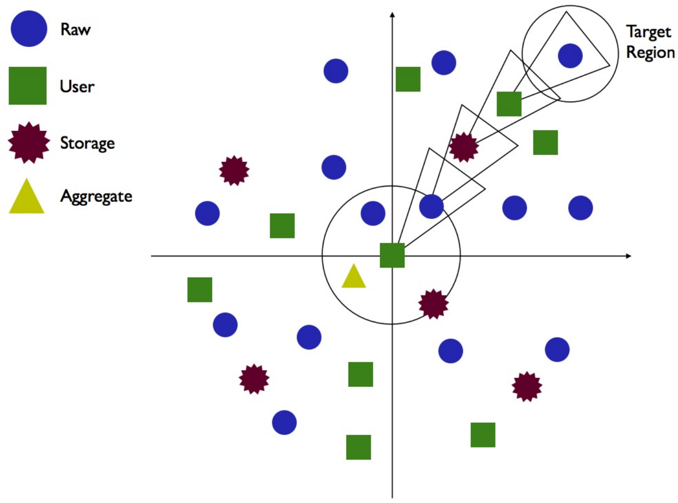


Figure 3.19: Propagation of search messages towards a specific region of interest.

aware of the searched data item. The angle of the search cone can be enlarged if the dimension of this list of nodes is not satisfactory. In this work, we consider $\alpha_{min} = 30°$ and $\alpha_{max} = 60°$.

[Search phase 2]

After a short time, the node that started the request propagation has a list of nodes that are aware of the data item of interest.

[Search phase 3]

In the most fortunate case, the node finds in the list the peer that generated the data item. From that peer, the node can obtain the list of storage nodes that contain the fragments of the data item. It is then sufficient to contact a subset of such storage nodes to re-build the original data item, as better discussed in Subsection 3.1.8.

If the publisher is not online, the searcher can collect a number of fragments from the storage nodes that have declared to be aware of the data item. Our system is robust against churn, since new fragments of the same data item are periodically generated (until the time validity of the data item expires) according to the strategies described in Subsection 3.1.8.

Network Coding Strategy

In the following, we detail the NC operations performed during the publication, retrieval, and maintenance of a given resource in the network.

Resource Publishing

A file of size $\mathscr{M}$, which needs to be stored, is divided into N_g generations composed of h fragments $\{s_i\}_{i=1}^h$ each, so that

$$\mathscr{M} = N_\mathrm{g} h d_\mathrm{F}$$

d_F being the size of each fragment.[1] We now focus on a single generation, since all the operations are the same for each generation. The fragments of a generation can be collected in a column vector denoted as $\boldsymbol{s} = (s_1, \dots, s_h)^T$, where $(\cdot)^T$ denotes the transpose operator. These fragments are symbols in the Galois field GF(q) and are linearly combined in order to obtain the following coded fragments:

$$\psi_j = \sum_{i=1}^{h} \beta_{ji} s_i \qquad j = 1, \dots, n$$

where β_{ji} are coefficients belonging to the same field. The redundant fragments can be collected in a column vector denoted as $\boldsymbol{\psi} = (\psi_1, \dots, \psi_n)^T$, so that the following matrix expression holds:

$$\boldsymbol{\psi} = \boldsymbol{B}\boldsymbol{s}$$

where $\boldsymbol{B}$ is a matrix whose rows $\{\boldsymbol{B}_j\}_{j=1}^n$ correspond to the coding vectors of the generated fragments $\{\psi_j\}_{j=1}^n$. In the presence of RNC, each coefficient β_{ji} ($j = 1, \dots, n$; $i = 1, \dots, k$) is uniformly chosen among all possible values in GF(q). This implies that there exists a non-zero probability that two coded packets are linearly dependent. However, it is well known that this probability is basically zero if q is sufficiently large [65].

[1] Since all fragments are supposed to have the same size, if a generation is composed by less than h fragments, zero padding is applied.

Each packet flowing in the network contains both the coded payload and a header, which carries information about the generation, which the fragment belongs to, and the global coding vector of dimension h, i.e., the coefficients representing the linear combination of the original symbols $\{s_1,\ldots,s_h\}$. If the generation has h elements in GF(2^m) (i.e., each element corresponds to m bits), the coding vector has size

$$\ell_{\mathrm{cv}} = m \times h \text{ bits.}$$

For instance, the coding vector for generations of $h = 50$ elements and $q = 2^{16}$ has size equal to

$$\ell_{\mathrm{cv}} = 16\,\text{bits} \times 50 = 100 \text{ Bytes} \tag{3.2}$$

and the overhead, with respect to an average payload, in our application, of 10 MB, is approximately equal to 0.001%. Therefore, one can conclude that the overhead is sufficiently small. In this work, we consider $h = 50$ and $n = 100$; similar results hold for other values of h and n.

Resource Retrieval

To retrieve a published file, it is necessary to obtain, for each generation, using the previously illustrated lookup functionality, h linearly independent fragments $\{\rho_1,\ldots,\rho_h\}$, which can be collected in the column vector $\boldsymbol{\rho} = (\rho_1,\ldots,\rho_h)^T$. Note that $\boldsymbol{\rho}$ is one of the possible subsets of size of h drawn from $\{\boldsymbol{\psi}_j\}_{j=1}^n$ ($n \geq h$). After these fragments have been collected, the following system of linear equations can be solved using the classical Gauss elimination algorithm [60]:

$$\boldsymbol{\rho} = \boldsymbol{A}\boldsymbol{s} \tag{3.3}$$

where $\boldsymbol{A}$ is a matrix whose rows $\{\boldsymbol{A}_i\}_{i=1}^h$ correspond to the coding vectors of the retrieved fragments $\{\rho_i\}_{i=1}^h$. In other words, $\boldsymbol{A}$ is a submatrix of $\boldsymbol{B}$ of size h. Therefore, the original symbols can be obtained as

$$\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{\rho}.$$

Note that in the presence of RNC, it is not guaranteed that $\boldsymbol{A}$ is full rank for every subset $\{\rho_i\}_{i=1}^h$ and, therefore, (3.3) has a unique solution. In this case, therefore, more fragments have to be collected until the matrix $\boldsymbol{A}$ has rank equal to h.

Resource Maintenance Strategy

The use of RNC can be taken into account in order to perform the following novel *proactive* resource maintenance strategy. Whenever a Client Node retrieves a resource, it generates a given number of new fragments for each generation, also checking for network dynamic condition evolution. In addition to this strategy, resource maintenance can be also carried out *reactively*, by periodically (i.e., every T_{M} seconds) checking the availability, in the Storage Nodes, of the resources. If, for each generation, the percentage of surviving fragments falls below a properly defined retrieval "guard threshold" (which represents the fraction of fragments below which the resource is likely to become soon unavailable), the Super Node generates new fragments independent of the surviving ones, and distributes them in Storage Nodes. The threshold is denoted as τ and is equal to a fraction of the total number of fragments, i.e., $\tau \triangleq \varepsilon n$, $\varepsilon \in [0.5, 1]$. The number of new generated fragments is chosen so that the overall number of available fragments for a generation is equal to n, i.e., the published number of fragments.

Note that, in the proposed system, proactive maintenance is performed only when a client has finished a successful download and, therefore, a resource may not be regularly maintained if it is not sufficiently "popular." However, the complexity of the maintenance operations can be reduced significantly with respect to the reactive approach. Moreover, as the number of exchanged control messages is significantly smaller, bandwidth waste is lower.

In the following, we will refer to the reactive strategy as "periodic maintenance" (PM), whereas the new proactive strategy will be denoted as "sporadic maintenance" (SM). Note that, in both cases, according to the taxonomy in [64], the maintenance strategy aims at performing *functional repair* and not *exact repair*. In both cases, in fact, the new generated fragments are not exactly the same as those lost by disconnected nodes, but they have the same characteristics.

3.1.9 Performance Evaluation

In this section, we discuss on the performance of the proposed architecture, that we have carried out by means of discrete event simulations. A sensor network deployed in the city of Parma has been simulated, considering n_{u} UN that move over realistic paths of length equal to $l = 100$ Km generated using Google Maps API. In this case, the user density is $\delta = n_{\mathrm{u}}/l$. Each simulated UN selects a different path and starts moving over it. Moreover, we have placed (with uniformly random spatial distribution) RDS and SN over all the map. ADS have not been considered, since they would not be different from RDS (in publishing data) and UN (in retrieving data).

The considered set-up is composed by 50 SN, 500 RDS, and different numbers of UN so that the linear density over the roads δ are equal to 10, 20, 30, and 40 nodes per Km. As in [73], each node covers a region of interest of size 19.6 Km^2 and a dynamic discovery period ranging from 1.5 min to 6 min depending on the number of discovered nodes. Ten hours of system life are covered. Within the first four hours, the overlay is initialized, whereas during the fifth hour, RDS publish data resources. From the sixth hour, UN start searching contents at random locations implemented using a Poisson process with a mean arrival of 10VT ($\simeq$40 seconds). Moreover, in this last period storage nodes are randomly disconnected, according to the following different scenarios.

1. Burst of SN disconnections during the sixth hour of the observed period. At the end of the sixth hour, the number of SN has been reduced to 10.

2. SN disconnections over the second half of the observed period. The final number of SN is 10 as in the first scenario.

3. Continuous disconnections and reconnections of SN modeled with a Poisson process with a mean arrival of 25VT corresponding to 1.5 minutes.

Simulation results have been averaged over three independent simulation runs in order to reduce statistical fluctuations. Longer simulations are actually under deployment.

Regarding data publication and search, the following performance metrics are of interest:

- average number of hops per publication;
- average number of exchanged messages for finding SN per publication;
- average number of hops per search;
- average number of exchanged messages per search.

The impact of coding, instead, has been evaluated through the following performance metrics:

- resource availability, defined as the probability that a given resource in the network can be reconstructed;
- average used storage space on SN.

Publication and Search

As described in section 3.1.8, the publication and search processes are distributed, involving all types of peers that participate in the DGT modeled in the simulations with a Poisson process with mean arrival equals to 10VT ($\simeq$40 seconds). For all simulated scenarios, due to DGT properties, all lookups were successful. Fig. 3.20 shows that the average number of message propagations (hops) to discover storage nodes for publishing or searching data as functions of δ. One should observe that these average number of hops are almost constant with respect to the density of UN. We remark that RDS are initially aware of few storage nodes, but when their knowledge increases over a threshold, they stop performing storage node lookups. The average number of publication hops is always less than the average number of lookup hops. The reason is that, in order to discover storage nodes for publishing data, the RDS considers the whole circular region surrounding itself. Instead, the search process covers a restricted region defined by angle α.

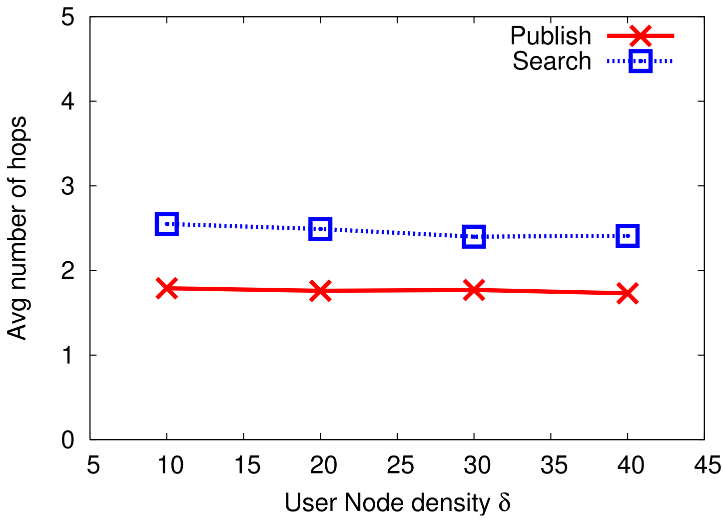


Figure 3.20: Average number of message propagations (hops) to discover storage nodes for publishing or searching data as functions of the UN density.

Fig. 3.21 illustrates the average number of exchanged messages for publication and search as functions of δ. The same considerations carried out in Fig. 3.20 for the average number of hops can be done in this case as well.

SN Disconnections

Fig. 3.22 shows that the resource availability, as a function of time, for the first scenario where there is a burst of SN disconnections. One can observe that, in correspondence of the disconnection burst, the resource availability reduces from 1 to 0.6. At this point, the availability starts growing again, since the maintenance process allows to introduce new fragments in substitution of the lost ones. As expected, the periodic strategy performs better than the sporadic one. This is due to the fact that in the latter case only popular resources are searched, whereas in the former case all resources (also those with minor popularity) are mantained. However, this comes at the price of a larger storage space occupancy on SN, as can be easily observed from Fig. 3.23.

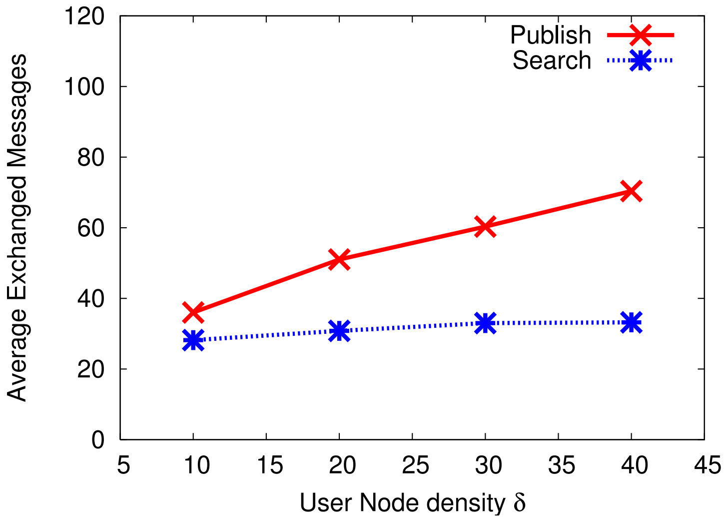


Figure 3.21: Average number of exchanged messages for publication and search as functions of the UN density.

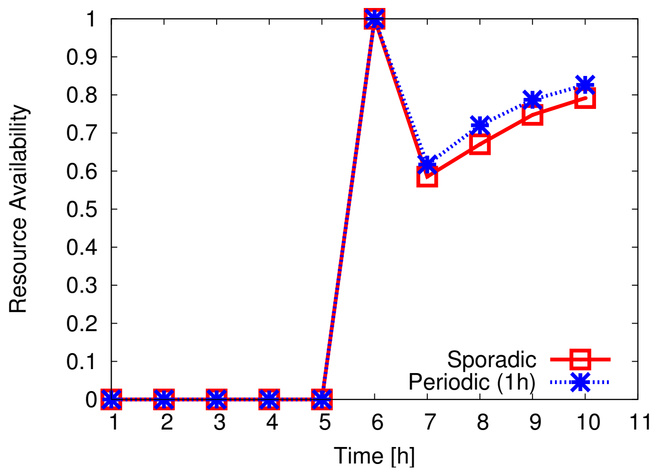


Figure 3.22: Resource availability, as a function of time, in the first scenario with a burst of SN disconnections.

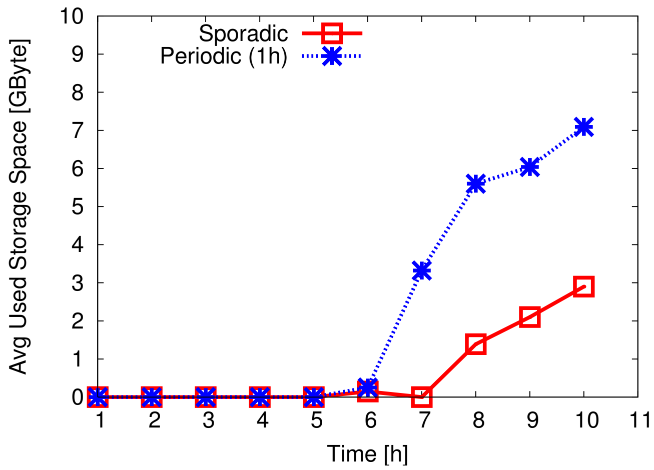


Figure 3.23: Average free storage space per node, as a function of time, in the first scenario with a burst of SN disconnections.

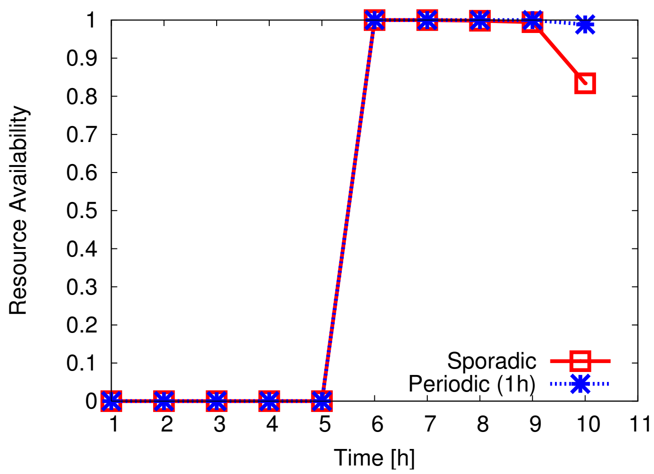


Figure 3.24: Resource availability, as a function of time, in the second scenario with SN disconnections over a long period of time.

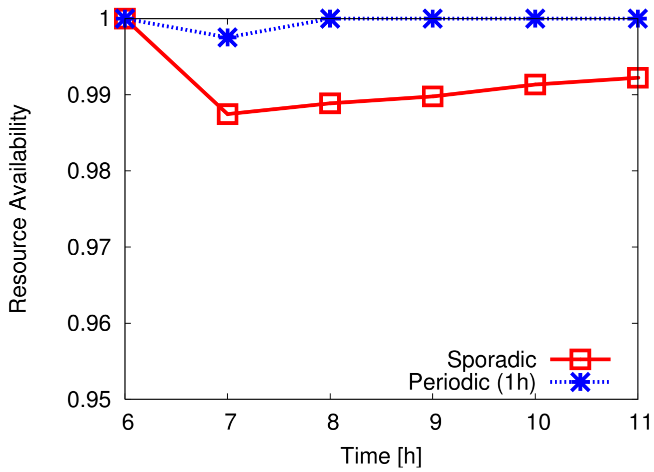


Figure 3.25: Resource availability, as a function of time, in the third scenario with continuous SN disconnections and reconnections.

In Fig. 3.24, the resource availability is shown, as a function of time, in the second scenario with SN disconnections over a long period of time. In this case as well, the maintenance processes preserve the resource availability to the maximum value (i.e., 1). In the presence of sporadic maintenance, however, the resource availability decreases during the last hour. The reason is that, the more the number of SN decreases, the higher the probability of file retrieval failure, which makes its reconstruction not possible.

Fig. 3.25 shows the resource availability, as a function of time, in the third scenario with continuous SN disconnections and reconnections. One should observe that with periodic maintenance the resource availability is not affected by such a churn process. Sporadic maintenance, instead, is not sufficient to preserve resource availability to its maximum value. The reason is that if resources cannot be retrieved, due to SN disconnections, new fragments are not inserted in the system.

Chapter 4

Conclusions

In this thesis, we have presented the design, implementation and evaluation of a new category of decentralized peer-to-peer (P2P) algorithms for location based services, called *Distributed Geographic Table* (DGT). The designed overlay allows each participant 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 proposed P2P scheme has aimed at overcoming the limits of proposed centralized and distributed solutions for *Location Based Services*. The following objectives have guided the design activity:

- to avoid that a central node or a particular peer could potentially 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.
- to reduce the risk of disseminating obsolete information in the network;
- to manage, by means of a distributed strategy, a high amount of simultaneous

updates and queries, limiting the platform cost and 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 have evaluated our scheme initially by means of the formal analysis of the proposed discovery procedure, and then by applying state of the art simulation techniques enabling a high level of realism for the modeled scenarios. We have shown that the algorithms achieve convincing performance for extended ranges of system parameters and are able to properly maintain the neighborhood knowledge of each peer, discover new nodes and updated information with low overhead in terms of exchanged messages. All parameters in our protocol can be tuned in order to achieve the most suitable trade-offs and performance for the considered application.

Furthermore, we investigated the performance of the DGT in two different appealing scenarios. Firstly, we used the designed overlay to build a distributed *Traffic Information System* (TIS) called "DGT for VANET" (D4V) for the dissemination of traffic alert messages. In such a system users can participate using their smartphone to send and receive real time information about traffic conditions or potentially dangerous situations. The D4V architecture has been evaluated using an extensive simulation phase based on established vehicular mobility models and through the deployment of a mixed network made of automatic D4V peers and a small number of real users with Android devices. The D4V prototype has been developed using the open source middleware designed in our laboratory, called sip2peer, and it was also tested on the PlanetLab testbed to verify network performance in a heterogeneous environment. Simulative and real measurements appear very close, thus confirming the reliability of the simulation implementation as well 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 coping reasonably well 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.

The second application combines the P2P-based DGT with network coding, to achieve large scale information management in smart cities. While the former technique emphasizes load balancing capabilities and avoids the presence of bottlenecks and single points of failure, the use of NC techniques brings redundancy and thus fault-tolerance to make information retrieval extremely reliable and applications like real-time streaming highly robust. Indeed, NC improves the performance of P2P content sharing systems since it mitigates the block transfer scheduling or piece selection problem, especially when nodes dynamically join/depart from the network. Moreover, NC is important also for another functionality of the smart city infrastructure, i.e., distributed storage: should a storage node fail, the stored information could be retrieved by properly combining the information contained in other storage nodes. Simulations of dynamic scenarios in a realistic environment have shown that the system is robust against storage node disconnections. Even in extreme conditions (e.g., 80% of storage nodes disconnected), maintenance strategies guarantee high resource availability. In particular, we have shown that periodic maintenance performs slightly better than sporadic maintenance, at the expense of much higher storage space consumption. On the other hand, sporadic maintenance is effective only with popular resources, since regeneration is performed only after a successful download.

4.1 Further Work

Multiple research opportunities may spark off from the work, analysis and results presented in this thesis. In this section we outline some of the most promising research directions.

First of all, we plan to investigate and take into account the estimation of peer trajectory (e.g. nodes traveling along highways), in order to reduce the number of exchanged messages. Then, an additional effort could be focused on the extension of the current model, by defining a new one that will combine the DGT overlay characteristics, user/vehicle mobility and connectivity/coverage type, thus providing a complete and integrated estimation of involved variables.

We intend to integrate local (ad-hoc networks) communication to directly retrieve available peers in the neighborhood and exchange useful information about GB data. The support for ad-hoc connectivity will be initially analyzed through simulative means to understand the new involved parameters as well as benefits and costs. Of course an extensive evaluation on the field will be mandatory to validate the choice of type of direct communication (Bluetooth, WiFi, NFC) that should be used, according to the application context — either vehicular or pedestrian, indoor or outdoor. Due to the characteristics of the DGT algorithm we believe that direct communication between two nodes with no infrastructure support seems feasible and potentially beneficial for protocol performance, allowing a node to communicate at the same time in multiple ways to its neighbors. This evolution will require only limited redesign and adaptation of data structures, in order to correctly identify the most effective method to communicate with a target node according to its distance, network interface or behavior/context.

Starting from the encouraging results obtained with our first D4V prototype, we also intend to organize and deploy a large scale experiment across the city of Parma, in order to evaluate the application with a huge number of active users over a long period. The experiment will provide us with multiple types of data and logs at different levels. Users will comment about the User Interface usability, allowing us to understand how to improve it to reduce the number of interaction between drivers and their smartphones. Network and DGT logs will provide the required information to analyze the performance in terms of data traffic requirements, coverage percentage and distance from the event. Last but not least ambitious development of the DGT/D4V solution will be the integration of the system with a smart vehicle able to sense the environment and automatically provide messages about external events or conditions. The communication with the smart vehicle engine will be obviously bidirectional, as the D4V node will be able to provide the car supervisor with information about environment conditions received from other vehicles, allowing it to adapt vehicle behavior or to reconfigure algorithms for the analysis of incoming raw data.

Acknowledgements

Finally, I am here watching the last white page of this editor as I am ready to write acknowledgements. Looking backward these three years were rich of activities, travels, papers and a lots of coding but essentially they have been full of amazing people that influenced and helped me during my work.

First of all, I would like to thank my supervisor Prof. Francesco Zanichelli and Dr. Michele Amoretti for supporting and guiding me during this long travel and at the same time giving me enough freedom to be independent in my research. I hope that it is only the beginning and that our research group may grow up and improve as we did during last years. I am also very grateful to Prof. Gianluigi Ferrari, Prof. Luca Veltri, Dr. Marco Martalò, Dr. Simone Cirani for our worthwhile collaboration and for our several useful chats that have driven my activities on the correct path.

Special thanks and deep gratitude go to Dr. Cecilia Mascolo for the big opportunity that she gave me to study and work for six months in Cambrige and in particular to be part since the first day of an amazing group of colleagues and friends. Many thanks to Christos, Ilias, Salvo, Tassos, Kiran, Liam, John, Haris and Jisun, I hope to collaborate again with you and look forward having a pint together one more time.

Finally a huge thanks to all my friends in Parma and Palermo. You are too much to be listed here, but I would like to say “Thank You” because you make me feel at home every day. You are always there for me even if we do not see each other for a long time.

This thesis is dedicated to my parents who have been by my side understanding my choices and supporting me all the time during these eight years of studying.

Grazie !

Marco

Bibliography

[1] A. Kovacevic, N. Liebau, R. Steinmetz, *Globase.KOM - A P2P Overlay for Fully Retrievable Location-Based Search* , Proceedings of the Seventh IEEE International Conference on Peer-to-Peer Computing, September 2007.

[2] S. Asaduzzaman, G.v. Bochmann, *GeoP2P: an Adaptive and Fault-tolerant Peer-to-peer Overlay for Location Based Search*, CoRR 2009.

[3] A. Harwood and E.Tanin *Hashing Spatial Content over Peer-to-Peer Networks*, In ATNAC, page 5, 2003

[4] Google Latitude. `www.google.com/intl/en_us/latitude/intro.html`.

[5] M. Amoretti, M. Agosti, F. Zanichelli, *DEUS: a Discrete Event Universal Simulator*, in Proc. of the 2nd ICST/ACM International Conference on Simulation Tools and Techniques (SIMUTools 2009), Roma, Italy, March 2009

[6] M. Li, W. Lee, and A. Sivasubramaniam. DPTree: A Bal- anced Tree Based Indexing Framework for Peer-to-Peer Sys- tems. In 14th IEEE ICNP, pages 12-21, Nov. 2006.

[7] D. A. Tran and T. Nguyen. Hierarchical Multidimensional Search in Peer-to-peer Networks. Computer Communications, 31:346-357, 2008.

[8] B.Namand, A.Sussman. DiST:fullydecentralizedindexing for querying distributed multidimensional datasets. In 20th IPDPS, page 10, Apr. 2006.

[9] E. Tanin, A. Harwood, and H. Samet. Using a Distributed Quadtree Index in Peer-to-peer Networks. The VLDB Jour- nal, 16(2):165-178, 2007.

[10] B. Liu, W. C. Lee, and D. L. Lee. Supporting Complex Multi-Dimensional Queries in P2P Systems. Jun. 2005.

[11] A. Guttman. R-Trees: A Dynamic Index Structure for Spa- tial Searching. In SIGMOD84, pages 47-57, Jun. 1984.

[12] Vodafone Machine to Machine. M2M. `http://m2m.vodafone.com/home/`.

[13] K. Aberer, L.O. Alima, A. Ghodsi, S. Girdzijauskas, S. Haridi, M. Hauswirth, *The essence of P2P: A reference architecture for overlay networks*, 5th IEEE Int.'l Conference on Peer-to-Peer Computing (P2P 2005), August-September 2005, Konstanz, Germany.

[14] M. Amoretti, *A Survey of Peer-to-Peer Overlay Schemes: Effectiveness, Efficiency and Security*, Recent Patents on Computer Science, pp. 195-213, Vol. 2, Issue 3, ISSN 1874-4796, Ed. Bentham Science, November 2009.

[15] J. Bronsted, L.M. Kristensen, *Specification and performance evaluation of two zone dissemination protocols for vehicular ad-hoc networks*, 39th Annual Simulation Symposium, Huntsville, Alabama, USA, April 2006.

[16] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M.B. Srivastava, *Participatory sensing*, WSWê06, Boulder, Colorado, USA, 2006.

[17] M. Caliskan, D. Graupner, M. Mauve, *Decentralized discovery of free parking places*, 3rd Int.'l Workshop on Vehicular Ad Hoc Networks, Los Angeles, California, USA, September 2006.

[18] M.P. Fewell, *Area of Common Overlap of Three Circles*, Tech. Report of the Department of Defence, Australian Government, October 2006, http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA463920

[19] M. Fiore, J. Harri, F. Filali, C. Bonnet, *Vehicular Mobility Simulation with VanetMobiSim*, SIMULATION, September 2009.

[20] T. Fujiki, M. Kirimura, T. Umedu, T. Higashino, *Efficient acquisition of local traffic information using inter-vehicle communication with queries*, 10th Int.'l IEEE Conference on Intelligent Transportation Systems (ITCS '07), Seattle, Washington, USA, December 2007.

[21] M. Gerla, L. Kleinrock *Vehicular networks and the future of the mobile internet*, Computer Networks, Vol. 55, no. 2: 457-469, February 2011.

[22] M. Granovetter, *The Strength of Weak Ties*, American Journal of Sociology, Vol. 78, No. 6, pp. 1360-1380, 1973.

[23] D. Hadaller, S. Keshav, T. Brecht, S. Agarwal, *Vehicular opportunistic communication under the microscope*, Int.'l Conference On Mobile Systems, Applications And Services (MobiSys), San Juan, Puerto Rico, June 2007.

[24] M. Han, S. Moon, Y. Lee, K. Jang, D. Lee, *Evaluation of VoIP quality over WiBro*, Passive and Active Measurement Conference (PAM), Cleveland, Ohio, USA, April 2008.

[25] J. Harri, F. Filali, C. Bonnet, *A framework for mobility models generation and its application to inter-vehicular networks*, 3rd IEEE Int.'l Workshop on Mobility Management andWireless Access (MobiWacê05), Cologne, Germany, August 2005.

[26] H. Hartenstein, K. Laberteaux, *VANET Vehicular Applications and Inter-Networking Technologies*, ISBN: 978-0-470-74056-9, December 2009.

[27] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, S. Madden, *Cartel: a distributed mobile sensor computing system*, 4th ACM Conference on Embedded Networked Sensor Systems (SenSys), Boulder, Colorado, USA, November 2006.

[28] R. Jedrzej, B. Scheuermann, M. Koegel, M. Mauve, *PeerTIS: a peer-to-peer traffic information system*. Int.'l Conference on Mobile Computing and Networking, Beijing, China, 2009.

[29] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, R. Herrtwich, *Design of 5.9 GHz DSRC-based vehicular safety communication*, IEEE Wireless Communications, 13 (5), 36?43, 2006.

[30] S. Krauss, P. Wagner, C. Gawron, *Metastable states in a microscopic model of traffic flow*, Physical Review E, 55: 55-97, 1997.

[31] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, A. Corradi, *MobEyes: smart mobs for urban monitoring with vehicular sensor networks*, IEEE Wireless Communications ,13 (5), 2006.

[32] U. Lee, M. Gerla, *A survey of urban vehicular sensing platforms*, Computer Networks, Vol. 54, no. 4: 527-544, March 2010.

[33] I. Leontiadis, C. Mascolo, *Opportunistic spatio-temporal dissemination system for vehicular networks*. Int.'l Conference On Mobile Systems, Applications And Services (MobiSys), San Juan, Puerto Rico, June 2007.

[34] A. Mahmoud, S. Olariu *Zipper: a zero-infrastructure peer-to-peer system for VANET*, Int.'l Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Chania, Crete Island, Greece, October 2007.

[35] P. Maymounkov, D. Mazieres, *Kademlia: A Peer-to-peer Information System Based on the XOR Metric*, First International Workshop on Peer-to-peer Systems, March 2002.

[36] P. Mohan, V. Padmanabhan, R. Ramjee, *Nericell: rich monitoring of road and traffic conditions using mobile smartphones*, 4th ACM Conference on Embedded Networked Sensor Systems (SenSys), Raleigh, North Carolina, USA, November 2008.

[37] M. Mun, S. Reddy, K. Shilton, N. Yau, P. Boda, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, *PEIR, the personal environmental impact report, as a platform for participatory sensing systems research*, Int.'l Conference On Mobile Systems, Applications And Services (MobiSys), Krakow, Poland, June 2009.

[38] A. Qureshi, J. Carlisle, J. Guttag, Tavarua, *Video streaming with WWAN striping*, Multimediaê06, Santa Barbara, California, USA, October 2006.

[39] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, M. Mauve, *Challenge: peers on wheels - a road to new traffic information systems*, 13th annual ACM Int.l conference on Mobile computing and networking (MobiCom), New York, New York, USA 2007.

[40] J. Santa, A. Moragon, A. F. Gomez-Skarmeta, *Experimental evaluation of a novel vehicular communication paradigm based on cellular networks*, IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, June 2008.

[41] I. Seskar, S. Marie, J. Holtzman, J. Wasserman, *Rate of location area updates in cellular systems*, IEEE Vehicular Technology Conference (VTCê92), Denver, Colorado, USA, May 1992.

[42] T. Shinkawa, T. Terauchi, T. Kitani, N. Shibata, K. Yasumoto, M. Ito, T. Higashino, *A technique for information sharing using inter-vehicle communication with message ferrying*, IEEE Int.'l Conference on Mobile Data Management (MDM '06), Nara, Japan, May 2006.

[43] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, *Broadcasting in VANET*. Workshop on Vehicular Ad Hoc Networks (Move'07), Montreal, Canada, September 2007.

[44] M. Trieber, A. Hennecke, D. Helbing, *Congested traffic states in empirical observations and microscopic simulations*, Physical Review E, 62(2): 1805?1824, 2000.

[45] A. Vakulenko, *Circles Intersection library for MATLAB*, http://www.mathworks.com/matlabcentral/fileexchange/5313

[46] A. Wegener, H. Hellbruck, S. Fischer, C. Schmidt, S. Fekete, *AutoCast: An adaptive data dissemination protocol fro traffic information systems*, 66th IEEE Vehicular Technology Conference (VTC '07), Dublin, Ireland, April 2007.

[47] H. Xu, M. Barth, *An adaptive dissemination mechanism form intervechicle communication-based decentralized traffic information system*, 9th Int.'l IEEE Conference on Intelligent Transportation Systems (ITCS '06), Toronto, Canada, September 2006.

[48] J. Zhao, G. Cao, *VADD: Vehicle-assisted data delivery in vehicular ad hoc networks*, IEEE Transaction on Vehicular Technology, 57(3), 1910-1922, 2007.

[49] B. Zhou, K. Xu, M. Gerla. *Group and swarm mobility models for ad hoc network scenarios using virtual tracks*, IEEE Military Communications Conference (MILCOM), Monterey, California, USA, October 2004.

[50] eMule project homepage, http://www.emule-project.net

[51] Sun Microsystems, Inc., *JXTA Technology: Creating Connected Communities*, White Paper, Sun Microsystems, 2005.

[52] Wi-Fi Alliance, *Wi-Fi Direct standard*, http://www.wi-fi.org/Wi-Fi_Direct.php

[53] E. Gustafsson, A. Jonsson, *Always Best Connected*, IEEE Wireless Communications, vol. 10, pp. 49-55, 2003.

[54] Z. Yan, H. Zhou, H. Zhang, and S. Zhang, *Speed-Based Probability- Driven Seamless Handover Scheme between WLAN and UMTS*, in MSN, Wuhan, China, Dec. 10-12, 2008.

[55] D. Kwak, J. Mo, and M. Kang, *Investigation of handoffs for IEEE 802.11 networks in vehicular environment*, in ICUFN, Hong Kong, June 7-9, 2009, pp. 89?94.

[56] J.A. Olivera et al., *VANBA: a simple handover mechanism for transparent, always-on V2V communications*, in Proc. on IEEE 69th Vehicular Technology Conference (VTC2009-Spring), April 26-29, 2009.

[57] F. Esposito, A. Vegni, I. Matta, A. Neri *On modeling speed-based vertical handovers in vehicular networks*, IEEE Global Telecommunications Conference (GLOBECOMÍ10), Miami, Fla, USA, 2010.

[58] S. Balasubramaniam and J. Indulska, *Vertical Handover Supporting Pervasive Computing in Future Wireless Networks*, Computer Communication Journal, vol. 27, no. 8, pp. 708?719, 2003.

[59] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, *Network information flow*, IEEE Trans. Inform. Theory, vol. 46, no. 4, pp. 1204–1216, July 2000.

[60] A. K. Atkinson, *An Introduction to Numerical Analysis*. New York, NY, USA: John Wiley & Sons, 1989.

[61] California Institute for Smart Communities, *Ten Steps to Becoming a Smart Community*, 2001.

[62] Distributed System Group, *Smart City videos*, http://dsg.ce.unipr.it/d4v

[63] Distributed Systems Group, *DEUS project web site*, http://code.google.com/p/deus/.

[64] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. O. Wainwright, and K. Ramachandran, *Network coding for distributed storage systems*, IEEE Trans. Inform. Theory, vol. 56, no. 9, pp. 4539–4551, September 2010.

[65] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*. Hanover, MA, USA: Now Publisher Foundations and Trends in Networking, 2007.

[66] C. Gkantsidis and P. Rodriguez, *Network coding for large scale content distribution*, Proc. IEEE Conf. on Computer Commun. (INFOCOM), Miami, FL, USA, March 2005, pp. 2235–2245.

[67] A. Kovacevic, N. Liebau, R. Steinmetz, *Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search*, IEEE Int.'l Conference on Peer-to-Peer Computing, Galway, Ireland, September 2007.

[68] M. Martalò, M. Picone, R. Bussandri, M. Amoretti, *A Pratical Network Coding Approach for Peer-to-Peer Distributed Storage*, Proc. of NetCod 2010, The 2010 IEEE International Symposium on Network Coding, Toronto, Canada, June 2010.

[69] G. Mega, A. Montresor, G. P. Picco, *Efficient Dissemination in Decentralized Social Networks*, IEEE Int.'l Conference on Peer-to-Peer Computing, Kyoto, Japan, August 2011.

[70] G. Montassier, T. Cholez, G. Doyen, R. Khatoun, I. Chrisment, O. Festor, *Content Pollution Quantification in Large P2P networks: a Measurement Study on KAD*, IEEE Int.'l Conference on Peer-to-Peer Computing, Kyoto, Japan, August 2011.

[71] M. Naphade, G. Banavar, C. Harrison, J. Paraszak, *Smarter Cities and Their Innovation Challenges*, IEEE Computer, Vol.44 No.6, June 2011.

[72] F. Oggier, A. Datta, *Byzantine Fault Tolerance of Regenerating Codes* IEEE Int.'l Conference on Peer-to-Peer Computing, Kyoto, Japan, August 2011.

[73] M. Picone, M. Amoretti, F. Zanichelli, *Proactive Neighbor Localization Based on Distributed Geographic Table*, Proc. of the 8th ACM-SIGMM International Conference on Advances in Mobile Computing & Multimedia (MoMM 2010), Paris, France, November 2010.

[74] M. Picone, M. Amoretti, F. Zanichelli, *Evaluating the Robustness of the DGT Approach for Smartphone-based Vehicular Networks*, 5th IEEE Workshop On User MObility and VEhicular Networks, Bonn, Germany, 4-7 October 2011.

[75] W. Rao, R. Vitenberg, S. Tarkoma *Towards Optimal Keyword-based Content Dissemination in DHT-based P2P Networks* IEEE Int.'l Conference on Peer-to-Peer Computing, Kyoto, Japan, August 2011.

[76] Smart Cities research group, MIT, *http://cities.media.mit.edu*, last access 17 September 2011.

[77] Distributed Systems Group, *Sip2Peer project home page*, http://code.google.com/p/sip2peer/.

[78] C. Fragouli and E. Soljanin, *Network Coding Applications*. Hanover, MA, USA: Now Publisher Foundations and Trends in Networking, 2007.

[79] MjSip Project Official Page: http://www.mjsip.org

[80] Steventon, A., and Wright, S. (eds), *Intelligent spaces: The application of pervasive ICT*, London, Springer 2006.

[81] United Nations, World Urbanization Prospects: The 2009 Revision - Highlights, 2010; *http://esa.un.org/unpd/wup/doc_highlights.htm*.

[82] Mobile Computing Wikipedia: *http://en.wikipedia.org/wiki/Mobile_computing*

[83] ISO 19132:2007 Geographic information – Location-based services – Reference model