Faculty of Computer Science and Mathematics

University of Passau, Germany

Context-Aware Mobility:

A Distributed Approach to Context Management

Amine Mohamed Houyou

Supervisor: Hermann de Meer

A thesis submitted for

Doctoral Degree

July 2009

1. Reviewer: Prof. Hermann de Meer
   Chair of Computer Networks & Communications
   Universität Passau
   Innstr. 43,
   94032 Passau
   Germany
   EMail: mailto:demeer@uni-passau.de
   Web: http://www.net.fim.uni-passau.de

2. Reviewer: Prof. David Hutchison
   Director of InfoLab21 and Professor of Computing
   Lancaster University
   LA1 4WA
   Lancaster, UK
   EMail: mailto:dh@comp.lancs.ac.uk
   Web: http://www.infolab21.lancs.ac.uk

Abstract

The recent development of a whole plethora of new wireless technologies, such as IEEE 802.11, IEEE 802.15, IEEE 802.16, UMTS, and more recently LTE, etc, has triggered several efforts to integrate these technologies in a converged world of transparent and ubiquitous wireless connectivity. Most of these technologies have evolved around a certain use case and with some user behaviour being assumed; however, there still lacks a holistic solution to adapt access to user needs, in an automatic and transparent manner.

One major problem that has to be addressed first, is mobility management between heterogeneous wireless networks. Current mobility management solutions mostly originate from cellular networking systems, which are operator specific, centralised, and focused on a single link technology. In order to deal with the wireless diversity of future wireless and mobile Internet, a new approach is needed.

Adaptive wireless connectivity that is tailored around the user needs and capabilities is named context-aware mobility management. Context refers to the information describing the surroundings of the user as well as his/her behaviour, and additional semantic information that could optimise the adaption process.

Context management normally entails discovering and tracking context, reasoning based on the discovered information, then adapting (or acting) upon the context-aware application or system. This context management chain is adapted throughout the thesis to the task of context-aware mobility management. The added complexity is necessary to adapt the ubiquitous access to the condition of both the user and the surrounding networks, while assuming that overlapping wireless networks could still be managed in separate management domains. Linking these management domains and aggregating

this composite information in the form of a network context is one of the major contributions of this work.

An overlay-based solution takes into account this scattered nature of the context management system, which is modelled as a decentralised dynamic location-based service. The proposed architecture is generalised to support ubiquitous location-based services, and a design methodology is proposed to ensure the localised impact of mobility-led context retrieval overhead.

To my late grandfather.

Acknowledgements

I would like to thank Prof. Hermann de Meer for introducing me to the world of research, for entrusting me with so many interesting and challenging tasks, and for offering me the possibility and support to work towards this thesis. I would also like to thank my colleagues at the University of Passau for creating an inspiring research environment, for the insightful discussions, and for helping with reviewing this work. I am also most thankful to Prof. David Hutchison for his thoroughness, his enriching comments, and for taking the time to be involved in discussing this work. My special thanks go to my parents, family, and friends for their boundless support, love, and encouragement that has always been there. I would especially like to thank my friends in Germany for making me feel at home.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Trends and Motivations for Research

The terms *"wireless"* networking and *"mobile"* Internet refer to two developments that both enable wireless access to the Internet. Wireless access refers to the last mile between the user and the access network being a wireless link. By contrast, the mobile Internet refers to the ability to provide wireless connectivity, while supporting terminal mobility, which results from the user's movement between different wireless base stations or access points.

The two fields of research, namely wireless networking and mobility support have witnessed considerable advances over the last two decades. On the one hand, Ethernet-like [25] *wireless local area network (WLAN)* has initially been developed to replace laid-out cables in office and home networks, by offering best effort packet-based wireless connectivity in a shared media. This could be considered the major development in enabling wireless networking. On the one hand, mobility has developed on a different track, i.e., cellular networks. The cellular networks have witnessed an even larger scale success than the fixed Internet. Currently half of the earth's population owns a mobile phone (according to the UN's *International Telecommunication Union (ITU)* [1]). Despite this success, a real wide spread of wireless Internet access through cellular networks has not yet been achieved. This is mostly due to the inflexible existing business and payment models, operating in a licensed spectrum. Furthermore, the cellular network architecture itself limits the ability to offer inexpensive wireless bandwidth, when compared with broadband WLAN and similar technologies.

1.1.1 The Limits of Existing Mobile Internet Architectures

Cellular networks have first been built with the purpose to support mobility for circuit-switched telephony services, which entails supporting service continuity and guarantee the reachability of the mobile user. The network allows deterministic resource reservation at a series of serving wireless access nodes along the movement path of a mobile user. The resulting technology is able to strictly mediate wireless resources for mobile users, where interactive real-time services and limited reconfiguration times can be guaranteed. However, this can only be achieved at the cost of complex mobility management procedures carried out for each user within the network with a limited bandwidth guarantee per user. On the other hand, Ethernet-like wireless techniques have focused on increasing bandwidth assigned to each user to almost approach broadband limits with the help of sophisticated modulation schemes and media access control mechanisms, while assuming very limited or no mobility. New attempts have been made to enable mobility management for the broadband type of wireless access. These efforts have resulted into a plethora of standards and techniques at both link[1] [167; 186] and network layers, which is IP-based, that can support mobility beyond the scope of a single local area network.

Nevertheless, IP networking architecture has not been designed to deal with mobility or with the variability of wireless links [26]. Instead, IP assumed a fixed user and end-host resulting into an address-centric paradigm, where IP addresses are used as both the identifiers of end hosts and routing directives or locators. Mobility, however, often refers to a change of the routing to reach the same end-host, whose identity stays the same. To deal with this new challenge, mobile IP has been developed in a way to mimic the mobility solution used in cellular networks. IP mobility [94; 98; 108; 140] offers only a limited interworking between heterogeneous wireless networks. In fact, mobile IP does not care why and how the connectivity service is maintained at the link layer especially in a rich spectrum, but mostly offers a re-direction service of IP traffic to the current location of the mobile user.

Instead of having a dominating wireless technique at the link layer and some universal mobility management system (at the network layer) [45], a number of technologies

[1] In the IEEE 802.11 link layer standard family alone, several new standards have been proposed such as IEEE 802.11f supporting roaming between different domains, IEEE 802.11p for vehicular networking, and IEEE 802.11r for fast roaming [2].

co-exist, still pushed by different applications or types of user activity or behaviour. Examples include that of cellular networks, which still offer the better alternative for fast moving users [126] and generally for interactive applications; whereas *digital video broadcasting for handhelds (DVB-H)* offers the better link technology for TV over IP downstreams and video distribution; while bandwidth demanding applications operate best in WLAN type of networks. This approach to mobility defines the principles of always-best-connected [40; 81; 89].

Even if all of the above mentioned applications can run on top of the same IP layer, IP on its own is unable to offer the intelligent wireless connectivity management that could support the selection of the wireless network that best suits user behaviour and application QoS needs. This requires a more cross-layered approach that takes into account several types of criteria in optimising mobile and wireless connectivity. An example of such an approach is context-aware mobility management [39; 88; 147].

1.1.2 Advancements in Handheld Technology

The assumption of a dumb end-host in cellular networks no longer holds, as more advancements in hardware miniaturisation and increased battery lifetime have allowed handhelds to gain on computing and memory power. The handhelds are also self-aware and location-aware (since the integrated *global positioning system (GPS)* chip [22; 95]). Other environment sensors are also integrated in the handhelds, allowing the possibility to offer context information for pervasive and context-aware applications. In addition to that, the advancements at the wireless interface capabilities are also driving the way to adaptive and context-aware mobile devices.

Context-aware applications require several processes running in a mobile computing, called a pervasive computing environment and communicating through different interfaces. The idea of using context to improve connectivity itself or the wireless services provided by a rich and diverse spectrum is still in its infancy.

The new developments at the mobile device already open the way to ubiquitously adapt to the spectrum diversity, where several technologies and modulation co-exist in the same frequency band. Both multiple antennas technology such as *multiple-in-multiple-out (MIMO)* [153], cognitive radios [115], and agile radios [123] are developed with the thought of enabling adaptivity even at the wireless physical interface. The

ability to program radio hardware and physical layer functionality makes it possible to build software defined radios [129] and software triggered radios.

These advancements have yet to be integrated in a mobility management architecture that can support the ubiquitous computing vision to take advantage of it. Separately to how the mobile device is capable of discovering the diversity and supporting it, it is meant to stay always-best-connected to meet the needs of roaming ubiquitous computing environment.

Ideally, a context-aware mobility architecture should support discovering the diversity without having to sense or probe the whole spectrum, continuously. It also has to adapt connectivity to application and service needs, while being able to take advantage of the self-awareness of the mobile devices to allow a better involvement of the mobile device in triggering wireless techniques on demand. This advanced connectivity service also suggests a better utilisation of what is called context to allow a semantically rich decision making, when adapting the connectivity. The concrete challenges addressed in this thesis are discussed next.

1.2 List of Addressed Challenges

Although the ideas of enabling context-aware mobility is not new [39; 91; 147], a holistic solution to integrate context-awareness in a mobility management architecture and an analysis of its implication is still lacking.

Context-awareness goes beyond the term *communication context* defined in RFC 3753 [120]. It is more of a description of the cross-layered condition of the application, the user, the user's behaviour, and his/her surrounding wireless diversity.

The problems addressed in this thesis include: *(1)* How to integrate context-awareness in a mobility architecture, with all the consequences and changes needed to that architecture? *(2)* How to deal with context heterogeneity and retrieve context in an information-centric manner? *(3)* What are the technological challenges to execute a context-aware or always best connectivity support on the communication path? More systematically, the challenges and problems addressed in this work can be described as follows:

1. Context-awareness is an essential topic of pervasive computing in general. The challenge is to identify the right context information that is suitable or essential

for mobility. Not all context information could be provided or retrieved in all systems, so there is an element of legacy which needs to be taken care of when designing a new context-aware architecture. The challenge is of identifying and modelling context.

2. The context-information collected in a mobile environment is distributed in nature. Both user and network sensors contribute to the construction of a reliable context-knowledge. The distributed management of such information could imply that different elements in the network track different parts of the context which need to be retrieved to allow a context-aware management decision. In other words, the system should retrieve context from where it has been generated/sensed.

3. Constructing a context-management system in a distributed system, needs to address the challenge of scalability. A networking approach to evaluate context-management is needed (i.e., communication overhead dependence on context description and level of decentralisation).

4. Use context-awareness to improve vertical handover between heterogeneous wireless domains. The handover requires a cross-layer knowledge including the handover procedure at each media or technology. This information could be seen as part of the context-information describing the network and should be retrievable when the user is as close as possible to the access network in question. Also, an atomisation of network domains needs to be defined. Where to draw the context boundary between wireless domains, and how much detail is needed in describing a wireless autonomous domain is really required.

1.3 Solution Approach and Thesis Structure

The integration of context-awareness is an attempt to deal with the heterogeneity and diversity at the wireless level. This means that the integration of the context-awareness has to be carried out taking different wireless and mobility systems into account, i.e., from cellular architecture to wireless networking, while representing this heterogeneity into a new context management system.

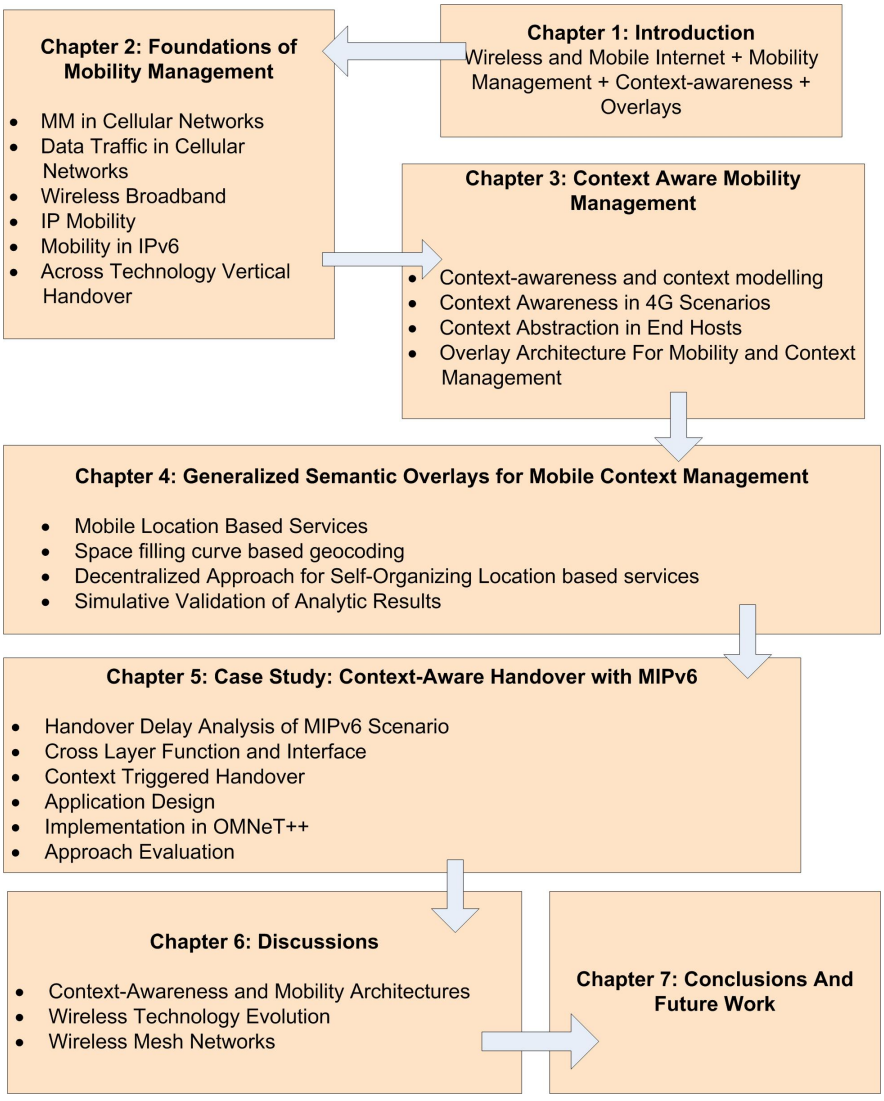

Figure 1.1: Thesis Structure - Block Diagram

The way this thesis is structured is shown in Figure 1.1. Chapter 2 looks into the foundations of mobility research and its evolution to suit the future of the wireless and mobile Internet. The ideas of dealing with wireless diversity and the existing architectural alternatives are investigated. The concept of context-aware mobility management is then introduced in Chapter 3. There, the context management chain that normally relies on sensing-reasoning-acting on context information is placed within the scope of mobility management. The context-aware mobility architecture, is centred on collecting context related to the user's changing situation. Therefore, alternative scalable information management systems suitable for mobile scenarios are discussed. A spatial semantic overlay is used to collect, cluster, and deliver the context information. Chapter 4 designs a peer-to-peer based solution to retrieve context information. The system design can be generalised to any mobile location-based services. A novel space model allows the design of a scalable information management system. While retrieving context information in a mobile environment is the major focus of this work, programmability of handover and context-triggered connectivity is explained in Chapter 5. The context-aware framework has to interact with lower layers to trigger handover between autonomous wireless domains. A simulative implementation of a context-aware connectivity triggering process demonstrates the ease of implementing such a system on top of existing technology with minimal or no change to lower layer mechanisms. Chapter 6 discusses the other architectural alternatives of context-aware mobility and carries out a critical analysis of the remaining open questions. Future work of the thesis and conclusions are covered in Chapter 7.

Chapter 2

Fundamentals of Mobility Management

The fundamentals of mobility management are visited in this chapter in order to understand the original problem domain. The evolution of mobility management in the cellular world has predated the introduction of proprietary solutions to support data-centered services in mobile environments and consequently the mobile Internet.

The evolution of both the mobility management protocols and that of radio access techniques is still continuing. The same could be said about the urgency for interoperability and interworking between these technologies, which is driven by the scarecity of spectrum. The existing proposals and their shortcomings are reviewed in this chapter.

Mobility management has been designed in cellular networks to guarantee mainly two things:

1. Tracking the user's location, so that placing a call to a mobile user is as fast as that in fixed telephony.

2. Supporting handover of circuit-switched paths to guarantee communication continuation despite movement.

Terminal mobility describes those mechanisms that redirect a given traffic flow to wherever a mobile terminal is (i.e., the traffic is directed on a fixed infrastructure from the caller until the current attachment point) [190, Ch. 5.4].

Mobility management in general refers to those mechanisms that guarantee terminal mobility. However, other types of mobility exist, such as personal mobility and session

mobility. But, only terminal mobility deals with the change in the wireless edge, which is the focus here.

In a cellular network, two mechanisms have been developed for this purpose; *location management (LM)* and *handover management (HM)*. Together they form the terminal mobility management solution.

2.1 Location Management

Definition 2.1.1 (Location management) *Location management, according to [29], is about designing two important tasks; Location registration and location update.*

Further functionality classified under location management is call redirection and call continuation, which have to be defined under the location registration and the location update.

Definition 2.1.2 (Location registration) *Location registration describes those mechanisms and algorithms used to store a location indicator or attribute in the network.*

Definition 2.1.3 (Call redirection) *Call redirection refers to the admission of new calls arriving from a* correspondent node CN *to reach a* mobile node MN[1]*. In general, a call or a protocol request is admitted by the network that manages the location of the user.*

Definition 2.1.4 (Location update) *Location update refers to the signalling efforts between the MN and different location trackers in the network that track the user's location changes and try to maintain a up-to-date information.*

Definition 2.1.5 (Call continuation) *Refers to the mechanisms used to make sure the communication between MN and CN can continue despite the location change.*

2.1.1 Location Management Algorithms

Location management (LM) has emerged as a key technique to enable mobile telephony [53; 78]. It involves storing the location of the user in location servers (found in the network), which could be queried by other network components each time a link or

[1] A mobile node is a term, which refers to the mobile terminal accessed the user. This term is used interchangeably with the term *"mobile user"* and *"mobile terminal"* throughout the thesis.

communication circuit needs to be constructed to reach the user [104]. In a naïve approach, the MN updates its location each time it transits to a different cell. Such an approach, however, would quickly overwhelm the network. Instead, several cells are grouped together in a so called *location area (LA)*. While moving within an LA, the MN does not need to update its location. Instead, the *location update (LU)* only needs to occur when the MN has left an LA to another one. When an incoming call arrives, the network routes the call to the last reported LA of the mobile terminal. To find out in which cell the user is best reachable, the network must page cells within the LA (also referred to as polling) to find the exact location of the user. The two basic mechanisms (i.e., location update and paging) consist the basic dynamics of LM and the main sources of signalling cost [104].

Increasing the size of the LA decreases the LU cost, while increasing that of paging. LM research has evolved over the last years to adapt the two mechanisms, while keeping up with the technological progress of the networking components, and the number of users the system has to deal with. Important to these efforts, are the assumptions on the user movement and the arrival rate of new calls. A user with rare calls does not require as frequent LU as a frequent phone user. To design a LM scheme, normally, mobility models and call arrival assumptions need to be taken into account. Mobility models specify the dynamics of user movement. Whereas, call arrival rates specify the need for LU. For instance for users with rare calls, it is enough to have rare LU. In fact, several possibilities of LU algorithms arise from the latter assumption and have resulted in static and dynamic mechanisms [104]. Examples of static LU mechanisms include:

1. Paging area based mechanisms, where the user only initiates a LU when moving between two LAs. For this, some detection mechanism of this crossing is needed at the link level.

2. Reporting cells mechanisms, which rely on fewer selected cells within a LA, where MNs have to carry an LU. Only when the user crosses to these special cells that an update is sent.

The disadvantage of the static update mechanisms is that they do not cater for different call arrival rates, and therefore, incurring a high paging cost (paging is then extended to several LAs to locate the user). On the other hand, dynamic update

mechanisms are adaptive to either individual user behaviour (especially movement) or to aggregate or groups of users. These mechanisms include:

1. Depending on the incoming call rate and mobility, the size of a MN's LA is determined.
2. Combining recent history of LUs to determine the user's movement direction in defining the paging area.

The LU in dynamic LM techniques have to be controlled and adapted through some control parameters, which include:

- Time-based approach, defines the update rate T (in seconds).
- Movement-based approach, which takes place every M cell crossings.
- Motion-based, which takes place every time the travelled distance exceeds a given distance D (in metres).

2.1.2 A Location Management Algorithm Example

This example demonstrates the way the different parameters of a LM are linked together. The example focuses on a static LU algorithm, which means that LAs are defined through the number of cells in a static way (i.e., once designed for a given area, they stay fixed). Now, assuming a uniform average velocity (e.g., assuming a maximum uniform velocity per urban area linked to street traffic velocity, pedestrian zone, or highway, etc.). The static solution offers an effective and an easy solution to physically dividing a network, providing the same LA to every user, without any customisation. The optimal static LA size algorithm, which uses a Fluid-Flow mobility model, states that in a network with uniform cell size, cell shape, and user movement speed, the ideal number of cells per LA is given in [78] as:

$$Optimal\ number\ of\ cells\ in\ a\ LA\ (N_{opt}) = \sqrt{\frac{velocity\ of\ the\ user \times location\ update\ cost}{\pi cell\ radius \times paging\ cost}}$$

$$N_{opt} = \sqrt{\frac{\nu C_{lu}}{\pi R C_{pg}}} \tag{2.1}$$

Where ν is the velocity of the user, which implies that larger LAs are more suitable for areas with faster moving users, whereas smaller LAs are more suitable for areas, where users are slower. R is the radius of the cell and C_{lu} is the LU cost, whereas C_{pg} is the cost of paging.

This equation states that for higher user speed and LU costs, a larger number of cells per LA is preferable, while a large cell radius and high paging costs imply that a small number of cells per LA is optimal. Obviously, users are not homogeneous, but with sufficient data collection and analysis of aggregate user movement patterns, Fluid-Flow is a relatively successful method to optimise static LM algorithms.

The total location management cost per hour, for a given location area, may be expressed formally, as in [107] as:

$$Total\ Cost\ =\ Location\ Update\ Cost + Paging\ Cost$$

$$C_{total} = C_{lu} + C_{pg} \tag{2.2}$$

This equation applies to the general case of location management, regardless of whether a static or dynamic scheme is used, or if the LM scheme is parameterised on a per-user or aggregate basis. The expansion of the C_{lu} and C_{pg} terms is dependent, however, on specific LM implementations. In [107], the paging cost given in dollars per hours is defined as follows:

$$Paging\ cost = \#\ of\ incoming\ calls\ per\ hour \times \#\ of cells\ in\ paging\ area \tag{2.3}$$

Whereas the location update cost is given as:

$$Location\ update\ cost = (\frac{user\ movement\ rate}{estimated\ moves\ to\ leave\ LA}) \times cost\ per\ LU \tag{2.4}$$

The rate of movement of the user divided by the estimated moves out of a given LA could be replaced in a more generalised form by the estimate of the residence time in a LA giving:

$$Location\ update\ cost = \frac{cost\ per\ LU}{estimate\ of\ the\ residence\ time\ in\ the\ current\ LA} \tag{2.5}$$

Besides the algorithms used, the infrastructure implied by the above algorithms is closely linked, which algorithms that can be used. Dynamic algorithms require a heavy computational cost, which can adapt any of the above parameters to dynamics of user behaviour. Fortunately, aggregation of users allows to limit the customisation of the algorithm per user. The problem with location tracking is that there is little knowledge about the intentions of the user to communicate or to change their movement pattern, by the network. Therefore, LU and paging are type of beacon based mechanisms, which try to sample the user behaviour through time. As it is known in sampling theory [84], the frequency at which a dynamic process needs to be sampled (through frequent measurement samples) requires a costly tracking or polling process, which should be faster than the dynamics of the measured process itself. The sampling frequency should change with changing dynamics. The tracking element should interpret the changes of the measured dynamic process and increase the sampling frequency with increasing dynamicity. In LM, the dynamic system is the mobile user, and the tracking elements are placed in the LM architecture. Next, the elements used in executing LM algorithms in cellular networks are looked at.

2.1.3 Location Management Architecture

Location updates are sent by the mobile node after physical link is established with the network infrastructure. The network stores the location information of each MN in a *location database*. The database entry of an MN is updated when the latter node *(i)* performs a location update or *(ii)* when the network performs a terminal paging during a call delivery to the mobile terminal. This database update procedure is called *location registration*. The data-base consists of *home location registers (HLRs)* and *visited location registers (VLRs)*. Along location information related to each user, an HLR contains services and other profile information. In each LA there is at least one VLR, which downloads location information and service information from the HLR. Each VLR is then connected to several *mobile switching centres (MSCs)*, which support MN handover and resource negotiation and allocation. When using an always-update LU algorithms (as is the case in *global system for mobile communications (GSM)*), then each move between two MSCs results into an update of the cell *identity (ID)* at the local VLR and requires an update of the stored MSC at the HLR. A move between two

LAs results in a change of VLR, which then requires an update at the HLR of both current VLR and current MSC entries.

In a call delivery, the area of registration of the MN is first queried at the HLR to find out the current VLR of the called user, then paging is used to poll cells within that region until the user is found. In 3G networks, there are another element of hierarchy named the *gateway location register (GLR)*, which is responsible for a *gateway location area (G-LA)*. Crossing the border of G-LAs leads to an update of the HLR, whereas changing VLRs is only reported to the GLR.

With the help of architecture elements, aggregation of cells in LAs can be done in a dynamic manner, which is the case for dynamic LM algorithms. Another use of the hierarchical structure of cellular networks, is the need to support path updates during communication within real-time limits. Indirection is more effective in a hierarchical infrastructure, where an attachment point change only requires a localised update. For this purpose, handover management is needed.

2.2 Handover Management

Definition 2.2.1 (Handover Management) *The handover management entails those mechanisms invoked to allow a MN to keep its connectivity with the network, while changing wireless attachment points. Generally, a handover process has to take care of the following challenges:*

- *Reduce the signalling and power overheads used to compensate for fading communication channels (e.g., through the use of power control).*
- *Reduce the worsening of QoS experienced by the mobile user, due to moving away from the attachment point, then due to lengthy discovery mechanisms and handover execution.*
- *Efficient use of network resources.*
- *Cater for scalability, high reliability (reduce blocking probability).*

Movement detection, admission control, addressing, and rerouting at the edge, belong to the task of handover [29].

Handover management normally leads to a renegotiation of wireless connectivity at each time the point of attachment needs to change. This process differs from one

wireless technology to other and is highly linked to the architecture used. It also involves several layers (when referring to the ISO-OSI layer model) to guarantee the continuation of communication. To better understand the differences between the types of handover, first, the mechanisms used in cellular networks are explained. There, a homogeneous network infrastructure is assumed, but the application needs are the strictest.

Cellular networks have developed the most strict and stringent handover mechanisms to support the strict requirements of voice communication, while being mobile. Channel fading, interferences, and resource limitations have all been integrated in the handover management. The whole renegotiation process has to take place within a small fraction of second, unnoticeable by human user, and of course, interruption-free.

2.2.1 Handover in Cellular Networks

Once on a call, the MN's change of location first leads to an increase of signal strength (i.e., power management) by both MN and *base station (BS)* to compensate for weakening signal. Other frequency channels are normally probed by the MN or scanned for any paging messages. This allows the MN to discover other neighbouring BSs, while measuring the signal quality received on each channel. The handover process could then be started leading to a location change (i.e., cell ID of attachment point). For this purpose handover management deals with the process of tracking the attachment point of a moving user, while on a call. As a result, not only the location servers are updated, but also the end-to-end circuit, which is admitted once changing the attachment point.

Cellular networks have attempted to make this transition unnoticeable by the user. For this reason, the resources allocated at the edge (i.e., within the visited cells) need to be negotiated within real-time limits. Once enough resources are available at the future BS, the handover can go ahead. Handover management could be split into the following phases: *(a)* the ability to detect movement away from the old cell and closer to the new cell (movement detection), *(b)* negotiating the resources at the new cell (admission control), *(c)* readjusting the circuit path, while guaranteeing the call continuity (binding update).

The mobility management in cellular networks assumes a dumb mobile device. This means that the mobile terminal attaches on the physical level to the BS once a trigger is received from the network. The MN is unaware of its location and, therefore, has to be paged or beaconed in order to allow the network to position it.

The signal strength indicating the proximity of the BS to the MN is interpreted at the cellular network level. Once this signal strength degrades, the network interprets that as movement. The MN informs the network of the signal-strength measurements to each possible BS. The network combines service needs with the SLA agreement with the user and the policy in order to decide, which BS the MN has to attach to, next.

2.2.2 Handover Initiation in Cellular Networks

To start a handover, the first task is to discover that the user is crossing the cell boundary to another cell. The process of detecting the appropriate crossing point is called handover initiation. The upcoming cell, however, has limited resources, therefore a negotiation whether to carry out the handover procedure or not is subject to admission control; if there is not enough channel capabilities to support the new MN without degrading existing channels, the handover request is rejected. In GSM, each wireless channel is carried by a different carrier frequency. This requires the radio transceiver of the MN to turn off the old channel before tuning to the new radio channel leading to an unavoidable interruption time. This is called a *hard handover*. In *code division multiple access (CDMA)* (used in UMTS), channels in neighbouring cells can use the same frequency since multiplexing of channels is carried out with the help of spread spectrum orthogonal codes, which coexist on the same frequency without interfering with each other. This allows the MN to receive several signals on the same frequency channel from different BSs. A handover, which occurs without requiring the old communication channel to be turned off before tuning to the new channel is called a *soft handover*. In both handoff schemes, handover initiation is usually based on the measurements of the pilot signal strengths received at the mobile. The mobile measures the received pilot signals from a candidate set of nearby BSs. The received pilot signal strengths are time averaged to remove the effect of fast multipath fading (i.e., reflections of the radio signals on buildings, for instance, leads to timely shifted copies of the same signal to interfere with each other at the MN). These averaged pilot signals are called *radio signal strength indication (RSSI)*.

The different techniques to determine the RSSI levels upon which the handover process should be started are reviewed in [134]. Common to all cellular networks, a handover is initiated before the *"receiver threshold"* power level is reached. The receiver threshold is the minimum RSSI level acceptable for call continuation at the old

cell. If the RSSI drops below the receiver threshold, the ongoing call is than dropped. If the upcoming cell is busy, the time interval between the handover request and the moment, where the receiver threshold is reached is used to allow some queuing delay for new calls to be admitted. Multiple thresholds could be assigned to different users dependent on their movement speed. A low-speed user spends more time in a handover zone, and therefore, is assigned a higher threshold (higher power means the handover occurs sooner than if lower power is tolerated). The high-speed users are assigned lower thresholds. Such algorithms with multiple threshold perform better with regards to forced terminations and blocking probabilities.

2.2.3 Handover Architecture

Similar to the location management, the handover architecture in cellular networks relies on network components that negotiate with the MN a handover decision. The main handover systems are:

Network Controlled Handover (NCHO) In first generation cellular systems such as the advanced mobile phone system (AMPS), mobile telephone switching offices (MTSO) handles the RSSI measurements and handover decision. The handover execution time is in the order of many seconds. The network elements suffered scalability problems [134].

Mobile Assisted Handover (MAHO) This refers to the changes introduced in GSM, where the RSSI measurement are carried out by the MN which then sends them to the BS periodically. The BS or mobile switching center (MSC) then decides when to handover. The handover execution time amounts of about 1 sec [134].

Mobile Controlled Handover (MCHO) In this type of handover the MN and BS carry measurements which are sent to the MN. Then the MN decides based on the gathered information on when to carry out the handover. Digital european cordless telephone (DECT) is an example of such a system, where the handover execution time amounts 100 - 500 msec.

A handover can also occur, not only between wireless attachment points (BS in the case of GSM), it also can take place between totally different network domains. The path of update might be higher up in the cellular system architecture, and might require

more sophisticated redirection mechanisms. In this case, two types of handovers can be identified inter- and intra- domain handovers.

Definition 2.2.2 (Intradomain handover or (horizontal handover)) *A horizontal handover refers to the process of link handover (mostly involving physical and link layers) when the signal from a BS cannot be received at the MN so that a switching to another BS with a better signal is necessary. This handover occurs inside the same network organisation (same provider) and within the same technology domain (from a GSM channel to a GSM channel). [29][190, S.87-88]*

Definition 2.2.3 (Inter-domain handover (vertical handover)) *A vertical handover occurs between heterogeneous networks. It usually refers to change of wireless technology such as the case of a GSM-to-UMTS handover. The vertical handover procedure leads to a drastic change of the network path and sometimes that of the MN's active wireless interface, incurring lengthy interruptions of the communication.*

A vertical handover can occur in the following scenarios:

- *When the user leaves the coverage zone of a certain technology (e.g., UMTS) to enter right after in a different network (e.g., GSM).*
- *When the user is already connected to a given network and decides to switch to a network coverage in their reach, which might offer other advantages (e.g., from a UMTS to* WLAN*).*
- *When network traffic is redirected to less loaded access networks, within reach of the user.*

[29][190, S.87-88]

In this work, vertical handover scenarios are assumed to be a precursor of the design of a context-aware handover management mechanism, which is independent of the link technology or architecture assumed behind each link technique. However, before moving to the main contribution of this thesis, it is important to understand the principles of handover management research and its evaluation techniques.

2.2.4 Movement Dependence

In order to deal with the different mobility patterns and behaviours users might have, a multilayer approach. Wireless cells are overlaid as micro and macro cells and the users

are assigned to each layer according to their speeds. The microcells cover about 500 m whereas the macrocells 35 km, for GSM900. Since the slow users are assigned to microcells and fast users to macrocells, the number of handovers can be considerably reduced. The movement and speed tracking is carried out according to the discrete RSSI measurements 6.1. In more advanced proposals, global coverage could be achieved by hierarchical cell structure (HCS) [71] adding to microcells picocells to refer to indoor wireless technologies such as WLAN. The challenges of mobility management in such hierarchical systems are discussed in [29].

2.2.4.1 Handover Frequency vs. System Degradation

A handover results in a rerouting of the communication flow to the new BS. So frequent handovers result in increased switching load on the network. However, if the handover occur too rarely, both the MN and BS are forced to increase their transmission power to compensate for the increased distance between the two. Besides the energy cost for a MN, this also results in an increased inter-channel interference, which reduces the quality of other neighbouring radio channels. These negative aspects of handover are called system degradation in [195]. The optimal handover algorithm is a tradeoff between the system degradation and the handover frequency. It is, however, difficult to give one exclusive metric to measure system degradation. Several proposals exist in that direction:

1. *Handover delay*, which is the time between when the handover should occur until the moment when it actually does.

2. *Crossover*, which is the distance the user moves from the cell boundary during the handover.

3. *Drop probability*, which refers to a call drop. This can occur when the beacon signal is below a given threshold so that the BS is considered out of reach.

4. Reciprocal of the *expected average signal strength (EASS)*, which is then mean value of the signal strength (averaged) received by the MN during its entire trajectory.

5. *Number of service failures*, where a failure occurs when the beacon signal is below the acceptable threshold even if the maximum signal power is used between the BS and the MN.

2.3 Data Traffic in Cellular Systems

Voice services aside, other data services have been integrated in cellular systems. The main service offered has been to allow IP traffic (which consists most of the data traffic of Internet services) in cellular systems. The problem is that of packing packet-switched bursty traffic in a circuit-switched technology. This introduced a matter of cost, since mobile networks not only provide circuits in the classical way used in circuit-switched telephony networks between end-users, but also track the circuits at the edge with a costly mobility management technique explained above. With the specification of the *general packet radio system (GPRS)* standard by the *European Telecommunications Standards Institute (ETSI)* [3] then by the 3^{rd} *generation partnership project (3GPP)* [4], packet switched communication has been enabled on top of circuit switched cellular networks like GSM and UMTS. The use of *time division multiple access (TDMA)* allows users inside a given cell to access the same uplink and downlink frequency channels and therefore share the same medium. Each user is then allowed access to the channel following TDMA multiplexing. The packets sent on those channels are bursty in nature.

Further developments to the GPRS early standard include the *enhanced GPRS* and *enhanced UMTS* based on the *enhanced data rates for GSM evolution (EDGE)* standard. In UMTS networks, a further development besides GPRS allows higher data rates, called *high-speed downlink packet access (HSDPA)* [5]. IP traffic is then tunnelled in these data links provided in cellular networks. The handover and location management, occur in similar way to voice communication. In both GPRS and UMTS the GPRS Mobility Management (GMM) protocol supports mobility management functionality [41].

The technology trend shown in Figure 2.1 illustrates the historic development of data encoding techniques and standards, and their evolution as the integrating technology of data traffic into the cellular world. The *fourth generation* of networks *(4G)* [6] refers to a number of developments at both the physical layer (e.g., *Flash-orthogonal frequency division multiplexing (F-OFDM)* and *multiple-in-multiple-out (MIMO)*); at the link layer (e.g., cooperative networking (through cooperative ARQ link layer protocols)

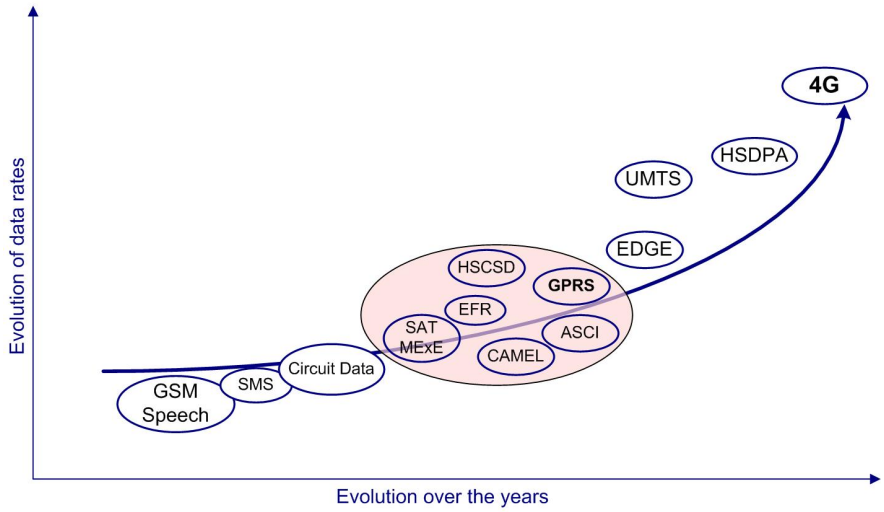

Figure 2.1: Cellular network evolution - Data traffic in cellular networks

and several IEEE proposals such as the media independence protocol IEEE 802.21); and at architectural level (e.g., 3GPP's *long term evolution (LTE)* [7]). These techniques should achieve transmission rates of between 100Mbps and 1Gbps. Today, in the 3G standard family up to 14Mbps are achievable in HSDPA. In fact, there is not yet one single definition or a standard to what is meant with 4G. However, common to the known 4G scenarios, is the all-IP world, which should be achieved.

The problem of terminology is again addressed in [67], which refers to the terminology used by ITU-R's *International Mobile Telecommunications (IMT)* standards and their evolution to that of the commonly used terms by the cellular industry. The latter defines the terminology of generations as follows:

- First generation (1G) - analog cellular radio (voice only)
- Second generation (2G) - digital cellular radio with some data services (SMS, WAP, then GPRS).
- Third generation (3G) - multimedia cellular radio (CDMA based) offering a full range of data services.

- Beyond 3G/fourth generation (B3G/4G) is still evolving and gathers advanced cellular technologies listed above and illustrated in Figure 2.1, interworking with other broadband technologies such as WiMAX and WiFi, based on an all IP-world such as the one envisaged by the EU project Daidalos [8].

The ITU-R's naming convention is centred around the evolution of the IMT standards, from IMT-2000 referring to the advances in the 2G systems (also called 2.5G and 3G), to Enhanced IMT-2000 (equivalent to 3.5G), and finally IMT-advanced which refers to 4G. The expected deployment of IMT-advanced is expected around year 2015 according to Eastwood et al. [67]. The latter researchers are active members of the IEEE 802.21 evolving standard, which offers generic description of link layer mobility functions while staying media independent (i.e., attempting to address any media).

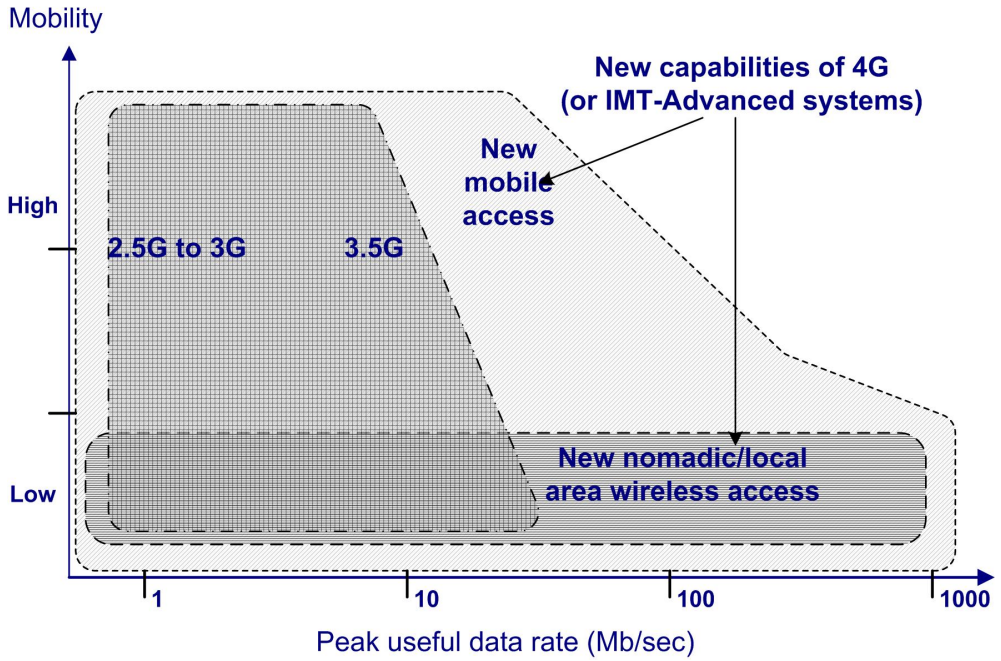

Figure 2.2: Mobility vs. 4G network technology - A differentiation of mobility per wireless technology

A further speculation about what 4G systems would look like could be summarised with the following words 2.3.1:

Definition 2.3.1 (4G Networks) *4G systems will be formed through a federation of both cellular architecture with wireless broadband technologies managed through an IP-*

based architecture. 4G will offer the mobile user a choice between highly reliable and guaranteed bandwidth via cellular link technologies, and between a number of IEEE standards which offer an alternative wireless technology suitable for more granular user behaviour and more coarse grained differentiation of application needs for reliability and bandwidth guarantees.

But first, the way data traffic is integrated in cellular networks and the impact of such services on mobility management is explained next.

2.3.1 Data Packet Network Architecture in Cellular Networks

Bettstetter et al. [41] present a comprehensive and detailed survey on how GPRS is deployed and how the different functionalities from mobility management to accounting and billing are carried out. UMTS follows the same architectural principle of GPRS, therefore, only those essential differences between the two technologies, which are related to mobility management are given here. In order to integrate GPRS in the GSM network, new architectural elements called the *GPRS support nodes (GSN)* have been introduced in [20].

A *serving GPRS support node* or (SGSN) deals with routing of packets, mobility management, and even *authentication, authorisation and accounting (AAA)* for those MNs within its service area. The SGSN stores user profile information (e.g., IP address, service class, etc.) and the current location of the MN (in the form of the current VLR and current cell).

A *gateway GPRS support node (GGSN)* acts as an access router (or gateway) between the GPRS network and the remaining of the Internet.

2.3.2 Mobility Management in GPRS Networks

Dealing with location management when using GPRS requires a different approach to location management methods used while accessing voice services. The GPRS traffic is iniated by the user and there is hardly need for call delivery mechanisms. Therefore, there is less need for strict location tracking. A MN can be in one of three states depending on its current traffic amount shown in Figure 2.3. This means both the LU update frequency, and thus, the amount of paging are defined for each state.

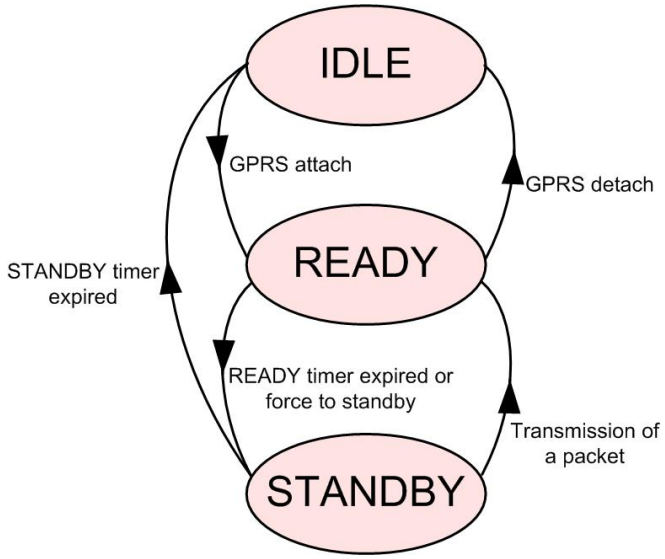

Figure 2.3: State model of a GPRS mobile node - Location update states

1. In the IDLE state the MN is not reachable, and no location update is performed by the MN. The network is unaware of the user's location. The MN can perform an *Attach* to leave this state and *Detach* to return to it.

2. Once the MN has performed an attach, the MN passes in the READY state. This state has a timer, which is reset each time the nodes communicates. The SGSN is informed each time a cell is changed. If a detach is passed, the MN falls back in the IDLE state. If the timer associated with this state expires, a state transition occurs to pass into the STANDBY state.

3. The STANDBY state is a transitional state, where the user might transmit more packets and then reenter the READY state. Otherwise, a STANDBY timeout is reached and the MN goes back to the IDLE state. The LU in the STANDBY state occurs once the MN moves between the so-called *routing areas (RAs)*. Each location area (LA) is divided into RAs, and each RA include several cells. Changing cells is not reported to the SGSN in this state. If the network had to forward some packets back to the MN, the exact cell is required. At this stage, paging is needed to locate the user.

GPRS mobility management consists of two levels. First, micro-mobility management tracks the current routing area or cell of the mobile station. This tracking is

performed by the SGSN. Second, macro-mobility management keeps track of the mobile station's current SGSN and stores it in the HLR, VLR, and GGSN [41].

2.3.3 Data Mobility in UMTS

Mobility management for data traffic is carried out in a UMTS network as follows. The location registration occurs in the HLR and in the MSC. For call forwarding, the SGSN and the GGSN are involved. The SGSN is responsible for packet forwarding to the right *radio network controller (RNC)*, which maintains a connection to the MN. The RNC monitors the delivery of packet to the user and maintains link-layer information about the user's connection to detect movement. The GGSN is the gateway to the Internet. For a location update, both the HLR and the MSC are informed of the location of the MN. In micro-mobility within the domain of an MSC, only an RNC is aware of this change, which does not involve the HLR nor the MSC. For call forwarding a soft-handover is used. In UMTS, there is no vertical handover to other network technologies (at least none which is anticipated in the UMTS network). When using the same operator, a MN can keep its IP address despite frequent handover [190, Ch. 2.5 - 2.7].

2.3.4 IP Traffic over GPRS

For IP-based application, GPRS is considered a link layer protocol on the wireless access part of the end-to-end IP path. As gateway, the GGSN can allow IP packets to be encapsulated in the GPRS datagrams. The MN can negotiate a dynamic IP address with a DHCP server serving all nodes within the operator's domain boundary. Each network operator has its own pool of IP addresses, which are allocated to communicating nodes. The problem with this approach, which is explained in [41] is its scalability. As an entry point from the Internet, in the general organisation, some GGSN has to then forward the datagrams to the current serving SGSN. Any IP address within the address pool the operator could be associated with a MN, therefore, Internet correspondent nodes are likely to use the same GGSN despite change of SGSN, increasing the load on certain GGSN nodes. As an alternative, some mobility management at the IP level could be placed to allow efficient routing and load balancing of state update due to mobility among GGSN nodes.

Besides GPRS, other wireless link technologies exist, including those that follow a totally different approach to cellular network architecture. These networks have been

introduced initially to offer personal or office-wide wireless connectivity, whose aim has been to replace wired *local area network (LAN)*, for instance. These technologies are, however, short ranged and rely on very rudimentary mobility mechanisms (if at all). They, however, offer much higher bandwidth (broadband) to a considerable number of users on, often, license-free spectrum (such as the case for IEEE 802.11 WLAN.

2.3.5 WLAN Infrastructure Mode

WLAN has been designed to offer a wireless extension to LAN. Mobility has not been designed in the original standards. The MN is equipped with a bootstrapping process (to discover and associate with a WLAN *access point (AP)* or other wireless nodes), which is the basis of a simplistic handover management. The MN scans the predefined WLAN channels for neighbouring APs. Once the MN moves out of reach of an AP the scanning process starts the scanning process again. If the two APs belong to the same IP domain, then, the horizontal handover only leads to an update of the LAN cache tables (i.e., using the *address resolution protocol (ARP)*). If the AP belongs to a different IP network, then a network mobility at the IP layer is needed in order to maintain communication session continuity. Mobile IP is used to make mobility transparent for application flows. In WLAN, the mobility management at the link layer is therefore rudimentary and depends on whether the old and new AP are somehow already associated with each other. If that is the case, negotiation of security association and assignment of wireless resources could be coordinated between several WLAN APs (for instance using IEEE 802.11f protocol) [162].

New wireless paradigms such as cooperative networking and *wireless mesh networks (WMNs)*, are among the technological leaps in the wireless research, which are very promising. One of the main goals in this research field is to increase the reliability of the wireless connectivity, while catering for more user dynamics and variability, such as mobility. These technologies should also easily interoperate with other mobile networks, which still offer the most reliable, but also expensive wireless network.

The type of integration envisaged for 4G networks include that of heavily using IEEE broadband standards like WLAN and WiMax, whenever possible.

2.4 Integrating Wireless Broadband in the Mobile World

Wireless broadband technologies have been mostly optimised to deal with providing high bandwidth in a shared wireless medium. The issue of mobility has been left to the network layer of the OSI model. This is especially the case for the most widely spread wireless broadband technique, WLAN.

Integration of different wireless technologies, depend not only on the ability to keep session continuation while moving between heterogeneous networks, but also to enable an architectural integration, where mobility management entities can still keep track of the user location or to deliver calls, when needed.

Therefore, there are two levels of integration; *(i) lower-level integration*, which includes the ability to carry out a vertical handover to a different link technology; and *(ii) higher-level integration*, where mobility management mechanisms should still be able to guarantee call delivery, wherever the user is located.

2.4.1 Lower Level Integration

The lower-level integration is concerned with managing wireless resources, where vertical handover between heterogeneous networks could occur. This challenge include all mechanisms needed to enable this type of handover at the MN side. This includes signalling and wireless link discovery mechanisms in heterogeneous systems as well as the execution of the handover procedure by the MN. Early proposals on inter-system roaming have developed overlay signalling using "hallo" messages (or beacons) sent at an overlay network that connects different APs. Therefore, detecting movement as it is done in WLAN can take place on a link-independent overlay proposed by Stemm and Katz [171]. Programmability of handover has been addressed in different ways evolving from programmable networks [109; 188]. There, a programmable network agent is sent to the MN, enlisting handover strategies and topological information. Handover policies defining, which wireless networks can influence the handover strategy that a programmable agent can include [188].

2.4.2 High-Level Architectural Integration

In addition to enabling vertical handover at the lower layers, an integration between two different architectural approaches to mobility is also a very challenging aspect.

The example of WLAN and 3G networks has been approached in several proposals [45; 80; 86; 179]. The two worlds are very different, since WLAN does not really cater for mobility management, whereas cellular systems do this in a very cumbersome manner. The two main approaches to this problem can be classified as *(i) tight integration*, *(ii) loose integration*.

Tight integration Refers to the approaches proposed around the late nineties, by adding WLAN APs to the cellular network, as an additional base station technology [69; 124; 180]. Handover is triggered according to the cellular network approach (mostly in a network assisted manner). The user's location is also managed according to the existing cellular architecture. The wireless APs are either integrated at the RNC level or at the GGSN. Once a MN is located, it is normally assigned to a normal UMTS cell, before the handover is triggered to the WLAN AP located inside the UMTS cell. This guarantees a tight resource management of the WLAN resources by the operator of a cellular network [161]).

Loose integration The all-IP-world is the other networking architecture, whose success led to the efforts to have an integrated architecture that supports mobility as well. Loose integration refers to the routing philosophy of IP. IP domains are loosely linked together since there are no permanent circuit relationships between access routers or gateways. Instead, with a small world topology, IP is able to connect any two IP nodes, within few hops. When applying this to wireless networks, WLAN or UMTS domains are seen as pure IP domains linked together through IP. If a MN hands over between the two networks, the only effort needed is in offering an indirection route to the user, in whichever network he/she is. In fact, this task is fulfilled by mobile IP, which tracks the user's location (current attachment IP domain) while supporting vertical handover. Examples of this approach all-IP solution [116; 179] have also been focusing on modifying the IP protocol to be adaptive to user behaviour and movement.

The first option can be seen as the most costly way to integrate WLAN, since wireless LAN is integrated closely as a further cell or extension to the UMTS cell. Load could be deviated through the WLAN AP instead of using UMTS, when this is needed. The management overhead combined with such a design consists in using the same overhead

required by UMTS to maintain location information. The UMTS network requires a network centric mobility management, where paging and constant probing are needed to detect movement, or discover the current location of the user. The incentive for a handover does not always take the user's preferences into account, but rather those of the operator, and might only be planned once the user is connected to the UMTS cell.

The second option requires a movement detection mechanism, is hard to achieve in a media-independent manner. In other words, first the WLAN AP has to be discovered by the user at the link layer, to then receive addressing information from the new gateway leading to a hard vertical handover of both the communication path over the Internet. For highly mobile users, this is not the most suitable form of handover, despite the almost zero cost related to mobility management. Furthermore, mobile IP has to be used to deal with the change of the Internet path.

It is worth noting the 3GPP's efforts to develop a new interoperability protocol layer (which could be mapped to the 2.5 OSI layer). This new protocol should be able to manage vertical handover (including link resource mediation and selection) in a media-independent manner. The IEEE 802.21 protocol offers an interoperability layer, which abstracts away from the details of the heterogeneous links, and should even allow to speed-up the handover procedure at the IP layer too.

2.4.3 IEEE 802.21 Media-Independent Handover

Media independence refers to a vertical handover, which is triggered and managed with a media-independent layer 2.5 add-on in the form of the IEEE 802.21 standard [9]. The current status of the standard is given in [67], which appeared in April 2008. In that paper, an example implementation based on IEEE 802.11 and IEEE 802.16 (WiMax) handover is thoroughly explained. In addition to the IEEE standardisation efforts, the IETF is also working on adapting transport and network layers functionality to suit the scenarios of IEEE 802.21. Integrating *digital video broadcasting standard for handhelds (DVB-H)* scenarios with the IEEE 802.21 has also been implemented and demonstrated within the European project Daidalos [8].

To summarise the standard, one could say that it is thought for a cooperative environment, where operators allow access to their network coverage data, which is used as an indicator of which link layer to beacon. There are three stages to this process *(i) handover initiation* including network discovery, network selection, and handover

Challenges in intertechnology handovers	Description	802.21-based solutions
Network discovery and selection	Handover decision making entity must continuously evaluate available access networks in a power-efficient manner.	IEEE 802.21 enables inter-RAT (radio access technology) network advertisements and also provides a mechanism to query candidate target networks (based on UE- User Equipment Ů location) and their properties.
IP session continuity	Minimising user disruption during handover requires session continuity when transitioning across radio technologies.	Requires L3 mobility signaling and a L3 anchor across networks. IEEE 802.21 provides link layer triggers that can optimise the performance of these L3 mobility management protocols.
Low latency and single radio transmission handovers	Minimise handover latency to support realtime multimedia applications. Coexistence and interference issues may mandate that only one radio in a dual-radio UE can transmit at a given time during handovers.	Both issues require target network preparation, while still connected to the source network. IEEE 802.21 provides signaling for resource query and resource reservation on the target network. This also requires inter-RAT interface between access gateways.
Operator control in target network selection	Operators may want to control/influence which target network the user selects. Thus, there is a need to support network-initiated handovers. This also requires reporting radio measurements (e.g., link signal strength) across different radio access networks.	IEEE 802.21 enables operators to enforce handover policies and decisions. A crucial issue for operators. Enables inter-RAT measurement reporting.

Table 2.1: Role of IEEE 802.21 in 4G, source [67]

negotiation. *(ii) handover preparation*, which includes triggering Layer 2 connectivity, and supporting IP connectivity. IEEE 802.21 helps with handover initiation, network selection, and interface activation. Regarding the context transfer and making sure that packets are forwarded to the right network is out of the scope of the IEEE 802.21 and is normally dealt with by the mobility networking architecture such as the case with IP mobility.

2.4.4 Movement Dependant IEEE 802.21 Handover Initiation

Looking at a concrete handover scenario, where a MN is initially travelling at higher speed, while connected to a WiMAX network (IEEE 802.16m). Suddenly, the user slows down. According to Eastwood et al. [67], some handover logic (most likely to be an application layer decision point) decides that connecting to IEEE 802.11 VHT (very

high transfer rate), would be more suitable. The MN uses the IEEE 802.21 service primitives to query a network based IEEE 802.21 handover information database. Once a response is received by the MN can select a network. It then can check the network using an 802.21 primitive called *"media independent command service" or (MICS*) to trigger the targeted interface to query the selected networks and obtain real-time status information. The MN can also carry out some pre-registration on the IP layer and related to AAA functionality. Up until this point of time, the communication and negotiation is carried out while still connected through the WiMAX interface. Once the right IEEE 802.11 VHT network is selected, and pre-registration has taken place, the MN uses another IEEE 802.21 primitive to switch on the IEEE 802.11 VHT interface and initiates the handover process.

IEEE 802.21 defines three services to support vertical handovers:

- *Media independent information service (MIIS)* provides topology information about neighbouring wireless networks as well as their properties and available services. This is the core of the work developed in this thesis, since this part of the IEEE 802.21 is not yet really standardised. This information has to be collected among heterogeneous network operators and relating to different technologies. According to Eastwood et al. [67], this is among the challenging problems facing a large scale deployment of IEEE 802.21.

- *Media independent command service (MICS)* to provide a way to manage and control link layer interfaces, as well as querying different access points or IP routers for information before the handover takes place.

- *Media independent event service (MIES)* to provide link layer triggers and link status events (such as link up and link down messages) and require a tight coupling with specific link layer messages.

To further complete the media-independent handover, packets and services need also to be transferred to the new access network with minimum effect on the application and the perceived quality of service (or quality of experience) noted by the user. This requires a networking service that allows mobility transparency such as mobile IP.

For a quick handover procedure at the IP layer, while integrating IEEE 802.21, the IETF MIP-SHOP group is defining a L3 protocol for carrying MIH payload and supporting 802.21 services at layer 3 between different access networks [94].

2.4.5 Open Issues in IEEE 802.21

Service and Network Discovery The approach in IEEE 802.21 suggests a database approach to store network information. This is recognized as a problem, since this database system has to be fed with information originating by different technologies and also requires a pre-agreed structure to scale for heterogeneous query frequency. The rich querying is also needed to satisfy user demands and capabilities. The IEEE 802.21 also mention the possibility of using XML based descriptions to cater for the heterogeneous management information stored and maintained in the database.

Network discovery at the Link layer Moon et al. [131] suggest using pure radio triggers to discover the wireless diversity and network coverage in a decentralised manner. In other words, each MN turns on its radio interfaces periodically to beacon each modulation domain separately. The authors of this proposal see this as a more scalable solution to discover the heterogenous wireless networks while saving having a tight power management to trigger each given wireless interface. The problem with this approach is that it is reactive, since it can only discover wireless coverage once the node has been in reach within reach of that network. The preregistration mechanisms and beaconing at each wireless network can lead to different results and is very dependent on the link technology. For instance in GSM/UMTS systems, the RSSI signals are sent continuously by the BSs and can lead to a fast detection of coverage, whereas in IEEE 802.11, this process normally takes up to few seconds for the MN just to beacon all possible channels and wait for a response from access points. Furthermore, discovering wireless network at the link layer does not take into account other query parameters such as pre-agreements or access rights, which also require some other level of negotiation.

Enhanced or intelligent handover triggers Ying et al. [196] explores other ways to trigger a handover, since a typical handover process is based on measurements and triggers from the link layer which ignore the needs of specific applications and that of user context. This work also attempts to integrate the dynamic tracking of network condition resulting in an enhanced media independent handover protocol (EMIH).

2.5 Loose Integration with Mobile IP

Inspired by cellular networks, *mobile IP (MIP)* has been introduced to allow mobility management at the IP-layer. In principle, a home registry, called the *home agent (HA)*, replaces the HLR and a *foreign agent (FA)* (introduced in mobile IPv4) the VLR. Mobile IP breaks down the end-to-end principle of the Internet, while using another property of the Internet, which is that global routing at the IP layer (i.e., delivering all packets between any two nodes in the Internet) can quickly be reestablished after localised changes of network structures, failures, or interruptions. The end-to-end principle, however, is not suitable for mobile environments since location and the address are tightly coupled together. A MN has to be readdressed each time its location changes, leading to the interruption of end-to-end protocol associations, which require the IP address of both end nodes, leading to an even more drastic change in the network than in cellular networks. This change of the IP address can be dealt with through indirection of traffic through the HA, with the assumption that the latter node, always has the most up-to-date information relating to the MN's current IP address. Other end nodes in the Internet can keep up protocol associations, with the mobile user by using the IP address of the HA instead of that of the MN. The HA plays the role of a proxy to the user, which is capable of redirecting traffic to the user's current IP address despite mobility.

2.5.1 Mobile IP Principles

IP terminal mobility can be split in two types of mobility scenarios, micro- and macro-mobility [190, Ch. 5.4.1].

Definition 2.5.1 (IP Micro-mobility) *Micro-mobility refers to a terminal mobility inside the same administrative IP-domain (i.e., the same autonomous system), which implies a smaller travelled distance before discovering a new attachment point. [190, Ch. 5.4.1].*

Definition 2.5.2 (IP Macro-mobility) *Macro-mobility refers to a terminal mobility between separate administrative IP-domains, which could be compared with vertical handover in cellular networks [190, Ch. 5.4.1].*

Most mobility management functionality in MIP is centred in the HA. The location management defines the current location of the MN as its current IP address, assigned

while attached to a given network, called the *care of address (CoA)*) [190, S. 153]. The HA has to store the current CoA of the MN. For the handover a signalling at the edge between the visited routers and the MN has been introduced in addition to traditional DHCP broadcasts. In MIPv4 a *foreign agent (FA)* takes the role of a proxy of MN and informs the HA of the new association [1]. In *mobile IPv6 (MIPv6)*, the MN informs the HA of its CoA directly. For the handover management, MIP relies on movement detection mechanism at the network layer, which, in its original proposal, is independent of any other layers. This means that the handover management relies only on layer 3 signalling and information exchange to deal with movement detection, network discovery, and attachment to new networks.

Mobile IP's main goal can be summarised as follows:

- Transparency for the application. Applications should continue communication despite change of CoA. One way to achieve this is by using the IP address of the HA (for instance in a TCP connection). The home agent transfers packets directly to the current CoA [190].

- Transparency towards the network. The IP routing updates do not affect any other elements in the Internet apart from the MN and the HA. The IP path between the two might change, due to changing CoA, but no other routing updates are needed anywhere else in the Internet [190].

- Fast vertical handover mechanism as soon as the CoA is known. The only update that has to occur is between the MN and the HA. The HA reports the new CoA as soon as obtained at the new network. This makes the handover delay depend a lot on the detection process of the CoA. The CoA could represent any type of heterogeneous wireless network organisation [190].

The historical introduction of MIP has faced limitations due to the drastic changes of the IPv4 protocol. Therefore, IP mobility has been part of the design process of the newer IPv6 protocol. Next, mobile IPv6 is thoroughly explained.

[1] This has been thought for backward compatibility, to make sure that the MN's IPv4 protocol should not be made aware of IP mobility protocol changes

2.5.2 Mobility in IPv6

IPv6 has been designed to improve on the shortcomings of the IPv4 protocol. Besides the considerable improvement of IPv6 address space from 2^{32} to 2^{128}, the IP header format has been simplified with additional extensions handled by extension headers instead of extending the basic IP header. This also has the advantage that the medium size of an IPv6 header is only twice as big as a former IPv4 header, keeping in mind, that at the same time the bit-size of IPv6 addresses has increased four times. Header size has been an important issue for forwarding and *checksum redundancy check (CRC)* especially for small-load packets, as is the case for voice-over-IP packets.

Furthermore, support for *Quality of Service (QoS)*, *IP-level security (IPSec)*, and efficient routing (improved fragmentation) have been included. Finally, IPv6 is designed to interoperate with the old IPv4 to provide seamless transition in the forthcoming years.

Another essential improvement to the IPv6 is its neighbour discovery protocol, which is a basis for movement detection as part of the handover process.

2.5.3 Neighbour Discovery for IPv6

The *Neighbour Discovery (ND)* protocol has evolved out of the IPv4's following protocols: *(i)* ICMP router discovery [58], *(ii)* ICMP redirect [144], and *(iii)* address resolution protocol (ARP) [143]. In addition to the above protocol functionality, a lot of new improvements have been implemented:

- Detection of failing routers, partially failing or partitioned links and nodes that change their link-layer addresses is now consistently performed through `neighbour unreachability detection (NUD)`.

- Prefix information can be used to achieve automatic address configuration; a separate mechanism to configure a network mask is made obsolete.

The two most important features of ND with respect to a handover consist in keeping track of neighbours reachability and in gathering information to perform autoconfiguration .

2.5.3.1 Auto-Configuration

The information gathered by a ND module at each IPv6 enabled node includes IP addresses and link-layer addresses of neighbouring nodes and available routers. Additional information sent by routers to end-user nodes include an IP prefix, used for auto address configuration, redirect information, and the length of the MTU.

2.5.3.2 Tracking Reachability

In order to keep track of the reachability of its neighbours, a node manages a neighbour cache with a reachability state for each of its neighbours. The corresponding finite state machine is defined in RFC 2461 [135] and has six states (see Table 2.2).

The state `INCOMPLETE` indicates a started but not yet completed address resolution. `Neighbour Solicitations` and `Neighbour Advertisements` are used to perform address resolution. The former requires a pull process sent by a node to its neighbours, and the latter refers to a push process where advertisements are received passively by the same node. If a neighbour is in the `REACHABLE` state, this indicates recent successful transmissions of packets. If no successful packets are sent for a certain time, `(ReachableTime)`, then the state changes to `STALE`. The `DELAY` state indicates that reachability is tested through upper layer mechanisms (such as TCP) but is not yet approved. `PROBE` indicates that reachability is now actively tested. And finally, `NULL` indicates, that the neighbour is no longer reachable. The above explained state transitions are further illustrated in Figure 2.4.

It is important to note, that only during the `PROBE` state, the ND is forced to actively check reachability. On the other hand, if no packets are sent for a certain time, unreachability is detected depending on passively listening for broadcasts sent by the access router behind each wireless link.

In the transitions between these states functions like *Router Discovery* and *Neighbour Unreachability Detection* came into play. Therefore handover happens.

The most important function to change the reachability state is probably the NUD state. Two sources of information are used to confirm reachability. The first are upper layer acknowledgments indicating successful transmission of a packet, which in the case of TCP would be in the form of acknowledgments. The other source of information is a received solicited *Neighbour Advertisements*, as they are only send in response

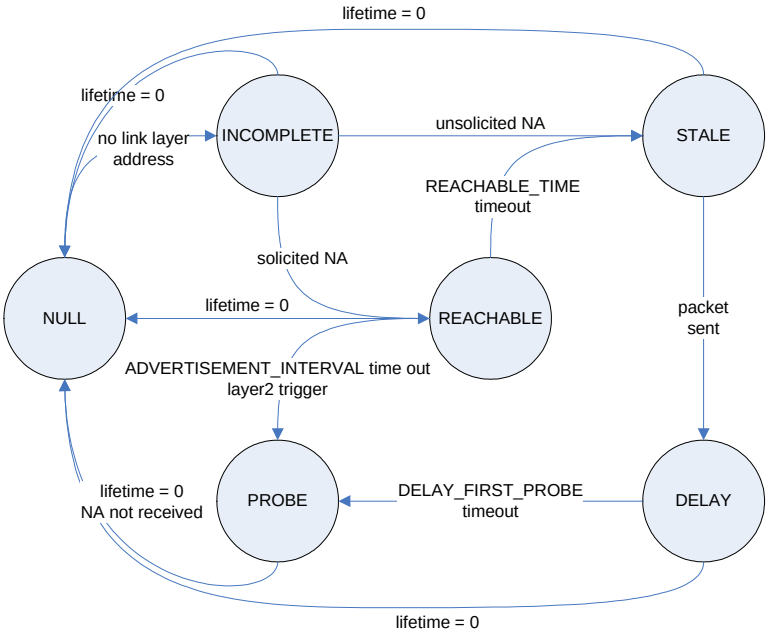

Figure 2.4: State transitions for neighbour discovery - Per direct neighbour state machine in each IPv6-enabled node

to a Neighbour Solicitation. This active confirmation is only performed, when the reachability state is in PROBE state, which is after upper layer reachability confirmation has failed. A confirmation of reachability both of upper layer or solicited Neighbour Advertisements change the reachability state back to REACHABLE. On the other hand if upper layer confirmation fails the state machine changes to PROBE and starts sending unicast Neighbour Solicitation messages every RetransTimer milliseconds for a period of MAX UNICAST SOLICIT. If this active scan also fails the state machine will transition to NULL.

The corresponding ICMPv6 messages needed to implement the functionality of ND are discussed next.

2.5.4 ICMPv6

ICMPv6 is designed to provide mechanisms to interchange control messages between IP devices. The two different types of messages defined by ICMPv6 are: error and informational messages.

Error messages are designed to indicate abnormal network behaviour. An example of

`INCOMPLETE`	indicates a started but not yet complete address resolution.
`REACHABLE`	within a certain time `"ReachableTime"` reachability has been approved.
`STALE`	For a certain period of time `"ReachableTime"` no reachability has been approved.
`DELAY`	If a packet is send while the node is in `STALE` mode, the node switches to `DELAY` giving upper layer protocols time `DELAY FIRST PROBE TIME` to provide a reachability confirmation.
`PROBE`	Actively sending `"Neighbour Solicitations"` every `"RetransTimer"` milliseconds until reachability confirmation is received.
`NULL`	After `MAX UNICAST SOLICIT` failed probes the nodes reachability state is set to `NULL`.

Table 2.2: States of the reachability state machine

Neighbour Unreachability Detection	determine whether a neighbouring node is still reachable.
Router Discovery	locate routers, which are attached on the same link.
Next-hop Determination	determine whether the destination node can be reached through the use of a router or is on the same link.

Table 2.3: Functions that change the reachability state

such an error message is the `Destination Unreachable` message that indicates a packet failed to deliver.
Informational messages on the other hand play a more important role in respect to handover. As stated before ND [135] is one of the RFCs that extend ICMPv6 by five more messages. The following list gives an overview of these messages:

- **Router Solicitation(RS)**: message indicates that a node requests a Router Advertisement from router.
- **Router Advertisement(RA)**: a router periodically advertises its presence to the attached nodes sending RAs. RAs also include additional link and Internet

parameters.

- **Neighbour Solicitation (NS)**: nodes use it to perform DAD and NUD as well as to determine the link layer address of its neighbours.
- **Neighbour Advertisement(NA)**: in response to a NS or to advertise a change in the link layer address.
- **Redirect**: message indicates, that a router points to a better first hop.

In addition to that, two base informational ICMPv6 message, `Echo Request` and `Echo Reply`, are used by the Ping6 application on application level in order to test reachability on the application layer.

2.5.5 Limitations of MIPv6 and Further Alternatives

MIPv6 is a macro-mobility solution which cannot deal properly with highly mobile users neither has it been built to support energy saving location updates. For high mobility, two problems exist:

- A handover initiation at the IP layer starts well after the handover procedure at the link layer has been completed, which is not suitable for fast moving users. The summed-up time in which a link loss (layer 2) is dealt-with and then the handover time required by the IP layer, together, consist the total handover delay. Any packets addressed to the MN that have been either generated at CNs or forwarded by the HA during that time are lost.
- A location change (i.e., reassociation with a new router and configuration of a new CoA) requires a state update at the home agent. Therefore, the update time is added to the time where the MN is not reachable.

For highly dynamic users, MIP results in a high traffic load, which is oblivious of the energy consumption profiles of mobile phones for instance. Such a state machine as developed for GPRS (2.3) does not exist for MIP.

In the last few years, several proposals have been made to address the problems faced for a large scale adoption of the MIP protocol. Intra-Domain mobility management protocol [128], mobile IPv4 regional registration [66], hierarchical mobile IPv6 [170],

fast handovers for Mobile IPv6 [102; 117], HAWAII [151; 152], and Cellular IP [46; 48] are among the main proposals. Common to all these approaches is their aim to limit the effect of location update on handover delay. Several reviews on the research needs in mobile Ip have been addressed in the past. The overview is based on a survey published by and that of Akyildiz et al. [29]. Reinbold et al. [158] provide the following criteria for comparing the different approaches.

Handover implementation How is handover implemented, what are expected delay and packet loss rates and how many stations are involved.

Adding paging to MIP If and how is passive connectivity supported [83], [199], [194].

Intra-Network traffic This refers to direct routing mechanisms inside a single domain instead of indirection (or triangulation) imposed by MIP. Relying on autonomous domains to quickly adapt to mobility through direct routing, can limit the update delays of MIP. The condition however is that the MN keeps the same CoA inside the IP domain [128]. Once a change of domain is needed, MIP is invoked leading to an update back to the HA.

Scalability does the proposed approach hold in terms of scalability.

In terms of routing optimisation Akyildiz et al. classify the different extensions to MIP as follows [29]:

- Tunnel-based schemes which use localised hierarchical registration to limit the update signalling and handover total latency. Creating regions as it has been done in *intra-domain mobility management protocol (IDMP)* [128] creates a hierarchical organisation of IP domains. This allows to separate micro-mobility domains that track *local CoA* changes from macro-mobility updates, which are reported back to the HA [170]and in *mobile IPv4 regional registration* [66].

- Routing-based schemes maintain host-specific routes in the routers. These routes are updated dependent on host mobility. *Cellular IP* [46; 48; 181] and *handoff aware wireless access Internet infrastructure (HAWAII)* [151], [152] are routing-based micro-mobility solutions. Routing-based refer to the cellular networking

approach of storing location attributes in a given infrastructure (VLR/HLR) and then having the system request this information (which a routing information) when a call delivery has to take place. These mechanisms cater for paging as well.

In terms of handover, the above protocols and proposals mostly rely on MIP layer handover triggers that use movement tracking and location sensing at the IP-layer (e.g., neighbour discovery and reachability state machine for MIPv6). Other methods exist to integrate location sensing at the link layer, so a SNR measurement at a IEEE 802.11 or a RSSI measurement below a given threshold in GSM/UMTS can be interpreted through cross layer machanisms as a *link drop* or *link down* for the host mobility detection mechanism. The host then can attempt to reattach to new visited routers more quickly. Fast handover in mobile IPv6 [108] specifies mechanisms that limit interruption times due to handover latency and their effect such as packet loss. Analytic approach to enhanced fast handover with low latency for mobile IPv6, where each functionality of the protocol and its necessity is investigated [117]. The new CoA is communicated back to the previous access router, where the handover could be started.

According to Zaidi et. al. [197], if the occurrence of a handover from one cell to another can be predicted ahead of time, the handover procedure can be initiated in advance. The use of movement prediction can be achieved quite accurately in cellular network [118; 195; 197]. These predictions can provide a *"handoff pretrigger"* incorporated in IP mobility protocols discussed in [82; 102], to provide transparent network layer mobility management.

The IP architecture is many way offers a natural inter-networking solution to integration of heterogeneous wireless networks. However, how cellular networks, which are intrinsically very complex, can be accessed in an all-IP world is addressed in Chapter 3. The idea of combining the HLR and mobile IP's home agent faces a scalability problem, as well as an architectural conflict. In the IP world, application and services are created and controlled in an end-to-end manner, whereas in the cellular world, networks are closely involved in creating and managing the service/application sessions. Combining the two architectures is not easy and requires a new architectural design that can deal just with mobility management parallel to both architectures. A promising new paradigm relies on a separation of management functionality on an overlay network

that can run independent of both architectures. Current state of the art on the role of overlays in mobility research is reviewed next.

2.6 Overlay Based Mobility Solutions

Overlays often refer to logical networks that exist on top of a different, sometimes, more complex underlaying network topology. Overlays often refer to a *peer-to-peer (P2P)* networks. The P2P paradigm is among the original paradigms assumed in designing the Internet. Networks are built by peering protocol entities that are equal in functionality. They operate similar routing protocols at the same layer.

Definition 2.6.1 (Overlay network) *Overlay networks often refer to the following two concepts:*

- *Applications, which communicate over logical links that differ or are independent from the physical path that is provided by the underlying network layer.*
- *Logical networks, which use the connectivity service provided by the underlaying topology, to construct a network organisation with its own routing and addressing schemes.*

[125]

Definition 2.6.2 (Peer-to-Peer) *P2P is a class of applications that takes advantage of resources storage, cycles, content, human presence Ũavailable at the edge of the Internet. Because accessing these decentralised resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers. [125]*

This definition is, however, that of P2P applications, which assumes a P2P relationship between the content provider and the content consumer in contrast to the client-server principle. But generally speaking, overlays can build a P2P network structure at any given layer.

Definition 2.6.3 (Overlay P2P networks) *Several nodes are said to have a P2P relationship if they do not need a central entity to self-organise and build a logical network based on local peering rules and algorithms, which exist at each node, equally at a peering layer. The emerging structure that results from these local rules are overlay networks.*

Overlay networks can support highly scalable and large distributed systems. This aspect has been one of the main reasons to adopt an overlay structure to replace the centralised home agent approach of MIP in the *robust overlay architecture for mobility (ROAM)* system [201], which is explained next.

2.6.1 Robust Overlay Architecture for Mobility (ROAM)

ROAM is an overlay-based solution for terminal mobility problem [201]. It offers both support for Internet routing, while replacing mobile IP infrastructure such as home agents and foreign agents and places this intelligence in an *Internet infrastructure indirection i3*-based network [173; 174]. Instead of having a home agent per user, location is tracked by a trigger for each user. An overlay node can manage several triggers simultaneously, while being independent of the managing node. In the case of the failure of the home agent in MIP, there is no built-in way to deal with this failure, whereas, i3 transparently shift or create triggers at any nodes in the overlay. ROAM has been designed to offer the following functionalities [201]:

Efficient routing Packets are routed using overlay nodes should achieve a latency smaller or equal to the shortest direct IP path.

Efficient handover Loss of a packet during handover should be as small as possible, and possibly should be avoided all together.

Fault tolerance Communication between two mobile hosts should not be more faulty than the communication between two stationary hosts.

Location privacy The location of the mobile user should not be retrievable by other end-hosts.

Simultaneous mobility End-hosts should be able to simultaneously be mobile and achieve a handover without leading to interruption of connection or communication session between the two.

Personal/session mobility A user should be able to move between devices while using the same service or application when a better possibility appears.

Link layer independence The user should be able to seamlessly switch between heterogeneous wireless link technologies, even if these link layers do not support the same type of mobility.

With ROAM, location change and handover lead to localised changes monitored by an overlay node. Which overlay node is assigned to which mobile user is less dependent on domain associations, but rather on overlay routing and load-balancing.

I3 suggests a rendez-vous point type of communication. Each packet is sent to an identifier instead of an end-host. To receive a packet, a node registers a trigger at an overlay node, and packets routed through the overlay network to the node which stores that identifier, will be forwarded to the addresses of the hosts associated with the triggers found. The trigger plays the role of indirection.

In the simplest form, i3 offers an API, which allows to generate packets in the form of $(id, data)$ where id is a m-bit identifier and $data$ is the payload in the form of an IP packet. A trigger takes the form of a pair $(id, addr)$, where the id is the trigger identifier, and the $addr$ is the IP address of the node associated with the latter trigger. A sender node sends packets to the id, as $(id, data)$ pairs forwarded by the overlay to the node that stores the trigger. The storing overlay node identifies the $addr$ of the receiver nodes, which have previously registered under the trigger with identifier id. Besides multicast, mobility can be addressed with this indirection, by allowing receiver nodes to update their current $addr$ with their current CoA.

The authors of ROAM [201] discuss the requirements set by a mobility management system and analyse the performance of ROAM in that aspect. In order to deal with legacy nodes, ROAM can coexist with other protocols such as mobile IP. Even in such scenarios the indirection proposed in ROAM offers a low latency stretch and is highly robust to node failure when compared with mobile IP.

The handover initiation is not addressed in ROAM, and handover is discovered passively by replacing the binding update to the home agent through an refresh process by the mobile host of its stored trigger. ROAM could be said to offer a good alternative to mobile IP when looking at robustness and scalability and latency of the binding process, while allowing a progressive and evolutionary change to mobility across different types of networks. It is also a good solution to integrate mobility management in overlay based networks, and for mobile services.

2.7 Chapter Summary

In this chapter, the problem of mobility management has been addressed in an evolutionary way. Looking at the early days of cellular networks, principles of micro- and macro- mobility have been defined. Also support for location management has laid the foundation for later proposals to support data and Internet traffic in mobile environments.

The networking principle of cellular networks offers a tight management of resources within the network to guarantee service delivery despite mobility. This comes, however, at a cost, which is introduced by managing the intelligence inside the network. An intelligent network suffers from the problem of scalability in the presence of IP-pushed services or applications, where new use cases such as gaming, video streaming, and P2P file sharing, are rather drivers of growth in the demand for more bandwidth, resources and flexibility. For instance, some applications or services might prefer a certain wireless network on others and might require a tight or loose management of the handover procedure and location tracking. This new level of differentiated wireless and mobility service is made possible through the coexistence of several wireless technologies and mobility architectures. Integrating all these technologies in a tightly coupled cellular network would cause the loss of the advantages behind several of these technologies. On the other hand, relying on IP mobility, although very powerful in offering a scalability and interoperability, is too agnostic to user movement or handover delay requirements.

Given the existence of different network architectures, overlay networks could offer a good abstraction to network organisations, while offering a new IP independent organisation and solution to mobility in general. This is the example of ROAM, which supports media independent mobility, while allowing support for legacy architectures.

Important to all these issues is the handover management between multiple technologies and the ability to discover application and user needs for mobility and the service type offered by a given network or link technology. This issue is called context-awareness and is addressed in the next chapter.

Chapter 3

Context Aware Mobility Management

In this chapter, two challenges are addressed, (*i*) the integration of context information in the mobility management of 4G wireless systems, and (*ii*) the enabling of context retrieval and management across heterogeneous network organisation, while taking mobility into account. The definition of context and context-awareness in mobile systems is looked at. In order to integrate context management in heterogeneous wireless networks, an overlay-based management system is proposed. This guarantees the creation of a scalable self-organising and robust context management system. The solution proposed does not preclude any networking architecture, neither does it suggest adding the whole complexity to an already complex system like a cellular network. In fact, the context management overlay composes between loosely integrated wireless networks, offering interoperability across operator and system boundaries.

3.1 What is Context Awareness

Early mention of the term has been made by the early 1990s [163]. For the pervasive computing research community, the notion of context is often related to context-aware computing and is commonly defined as the characteristics of the computing environment [38]. Context-awareness is sometimes defined through examples or synonyms of the word context [75]. For instance, context is referred to by [163] as location, identities of nearby people and objects, and changes to those objects. The more formally and widely

accepted definition of context (3.1.1) is given by Dey and Abowd [24; 62].

Definition 3.1.1 (Context Definition) *"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [62].*

Another similar definition is given by Chen and Kotz [51].

Definition 3.1.2 (Context Definition) *"... the set of environmental states and settings that either determines an application's behaviour or in which an application event occurs and is interesting to the user." [51].*

The states and settings can vary in their significance for a context-aware application. An environmental information or setting is only important if it can be viably used by the computing environment or application. As an example, location is often mentioned as an important context dimension. In location-aware systems [51], the computing environment attempts to make applications take advantage of mobility knowledge rather than trying to hide it. Context is, however, more than just location. It is also the environmental characteristic like temperature, light, or noise levels, which can be integrated in the computing system through physical sensors and help in adapting context-aware applications to changing environmental conditions. This view of how to use context is mostly centred on building applications, which are more effective and adaptive to the user's needs and situation without consuming too much of the user's attention [51]. Adaptiveness is an important aspect of context-awareness, and therefore sensing and collecting context has to be carried out in a dynamic and continuous manner.

How context is collected and used by applications depends greatly on a general taxonomy of context-aware computing systems. For instance, Pascoe et al. [138] propose a taxonomy that defines context-awareness as being able to sense context, to adapt to its changes, to discover new context resources, and to augment context [1].

Context-awareness has often been linked to adaptivity of applications to the user's context. In mobile environments, this takes another dimension since due to personal and host mobility, the environment can change frequently, during the lifetime of a service session.

[1] Context augmentation is about linking digital data like temperature, day of light brightness, to context information that is machine readable [51]

As an example, rendering a video stream on a mobile phone differs in the application context from when the user watches the same video stream on a laptop. The context information "screen size" and "mobile user" can be sent to the video server to then adapt the server process to send the right format of a video, best suited to each of context. Another example of context-aware applications are location-based services, which tailor the accessed service to the location of the user at a given point in time. In the latter example, the location is one of the context dimensions needed to support such a service. The other essential dimension is time.

In mobile cellular networks, this type of context information has to be retrieved from a wide number of sources, such as user profiles, location systems, and traffic monitors [145]. Dey [62] states that certain dimensions of context are more important than others. The dimensions that best represent the situation of an entity or a user are *location, identity, time, and activity* [62].

Most of the work on context-awareness concentrated on application or service context-awareness. But context can be used at system level too. The video application example, assumes certain fixed characteristics of the mobile network (e.g., higher probability of session interruptions, limited bandwidth, and fading connectivity, etc.), and adapts its encoding technique to the changing environment. In the approach pursued in this chapter, the question of how to control the network based on context information. So the adaptation can involve both application and network to maximise the user experience.

Prehofer mentions in [145] that context-awareness can be used to enhance network functions such as mobility support. Both user and network situation is used to adapt connectivity between networks and users. The process of mobility management becomes that of context-aware connectivity adaptation. A major challenge in building a context-aware mobility management system lies in describing and tracking context information in heterogeneous and dynamic environments, where both the situation of users and networks can change. In addition to that, integrating context-awareness at system level can be a complex task and some atomisation of the system representation or of its components has to be done beforehand.

3.1.1 From Entity Context to Network Context

Entity-based context refers to context information, which is normally generated or centred around a given entity such as the user, or the application. According to Schmidt et

al. [165] entity-based context is also easier to capture and describe than a system-based context.

Entity Context "*The domain knowledge about a specific entity is more universal and easier to establish than the domain knowledge of a complex system, and hence it is simpler to identify and implement contexts on entity level than on system level*" [165].

The key to integrating context in systems is to achieve some atomisation of the system as a whole or of some of its components into describable entities. Mobile networks are examples of complex systems, whose main goal is to provide connectivity to mobile users. Making these systems context-aware, can be defined as adding intelligence about the user and application context to the process of providing connectivity. The resulting network is context-aware since its connectivity service is adaptive to context information.

3.1.2 Context-Aware Networks

Recently, several initiatives have also gone towards defining context-aware networks. For example, Mathieu et al. [121], Giaffreda et al. [77], and Ocampo [137] have all addressed context-aware networks to some extent. Raz et al. [157] define context-aware networks as follows.

> "*A context aware network is a network that tries to overcome the limitations of the dumb and intelligent network models and to create a synthesis which combines the best of both network models. It is designed to allow for customization and application creation while at the same time ensuring that application operation is compatible not just with the preferences of the individual user but with the expressed preferences of the enterprise or other collectivity which owns the network.*" [157].

A dumb network relies mostly on intelligent end-systems (e.g., PCs and server machines) that create and manage applications and services. The network itself does not control or monitor application creation and operation. The dumb network delivers traffic associated with edge applications with no distinction between the transported flows. It offers a communication service which is the same to all types of application. It is then

up to the application to work with the offered resources and even adapt to the condition of the network. This is called the end-to-end principle. A well known example of such networks is the Internet.

In contrast, an intelligent network keeps a tight control over the creation and running of services offered to the users of the network. Reliability and availability play a major role in intelligent networks. An example would be telephony and broadcasting networks. Raz et al. [157] argue that the synthesis between the two types of architectures could be in the form of context-aware networks. For Raz et al., context-aware networks already exist in the form of the semantic Web, Grid networks, pervasive networks, autonomic networks, application-aware networks, service oriented networks, which all contain elements of context-awareness [157].

Under this view, a mobile cellular network could be said to be an intelligent network. Services are been built by networks operators and the network is dimensioned and architected to track the user movement, device capabilities, and other personal context information. However, when accessing Internet applications and services, the end-to-end principle adds an additional control mechanism which is unaware of the network intelligence.

Extending an intelligent cellular network to track IP-based applications is also questionable. The lessons can be taken from the integrated services QoS architecture (IntServ) [176], which faced scalability problems when forcing each network element to negotiate resources allocated for each microflow. The IntServ architecture never succeeded in replacing the best-effort service of IP, because the networking layer became too complex to function correctly.

Even if cellular networks have been designed to track the exact behaviour for each single user, this proves to be unnecessary for most IP-applications. Instead, other network architectures exist to provide an alternative wireless connectivity service, such as the case with *wireless mesh networks (WMN)* [28]. Context-awareness can be defined as the process to intelligently select between different networking and mobility architectures depending on user and application context.

The way to approach this is by defining wireless network entities representing connectivity islands, which offer different connectivity guarantees or networking services. The decision process of when and how to switch between the two types of networks is left to a context-aware mobility management system, designed in this work. The way

network context differs from information gathered by modern network management systems is not yet clear [137], neither is the granularity of the information gathered to describe network entities. The approach proposed in this work aims at an atomisation of access networks, which are seen as entities describable through their context. These descriptions define a clear interface to interact with access networks, while hiding the management of the actual infrastructure and inner topology consisting the network entity.

Composition of several context systems is also key to a scalable large scale solution. To describe an access network as an entity context, a schema or context model has to be agreed upon, which is then used to generate a description of an access network generated at a management point specialised in tracking the network context locally for the modelled autonomous system. Discovering and mediating context requires a separate management infrastructure, connecting the local management points and resulting into a *context-management system*.

3.1.3 Modelling Context

A context model is needed to represent context in a way that makes it machine readable. The context model is partly constrained by what context is used for. Whether the context-awareness is only used at application level (to improve the quality of experience for example) or at system level, a context model defines the way how to describe, store, and interact with context information. More details on existing context models are thoroughly explained in [38; 51; 137; 177]. The main classes of known models are:

- Key-value models: where the capabilities of a service or an entity are described with $(key, value)$ pairs, where keys are simpler identifiers of a given context dimension (like location, time, entity identity, etc), and whose value can be defined through an attribute value *value* associated with a some *key*. As an example in overlay networks, content is represented through a $(key, value)$ pair, where the *value* represents the semantic identity component of the content (like file name, or the IP address of where it is physically stored), and the *key* represents the overlay ID, which is used by the overlay to route or to retrieve the content.

- Markup scheme models: attempt to describe context with a standardised or pre-agreed hierarchical data structure. As a result, profiles extend the entity-attribute-value into hierarchies of one or more entity-attribute-value models. The markup schemes define a dialect based on XML which is used to describe context. Examples include Composite Capabilities/Preference Profile (CC/PP) proposed by the W3C [105].

- Graphical models: UML graphical diagrams can also be used to describe and model context. They are mostly human readable relationship diagrams between entity context.

- Object oriented models: use one or more of the properties of object oriented models to represent entity context (e.g., encapsulation, reusability, inheritance, class methods and messages).

- Logic based models: have a high degree of formality. Facts, expressions, and rules are used to define a context model. Such a model allows inference (or reasoning) to derive new facts based on existing rules in the system.

- Ontology based models: ontologies represent a description of the context concepts, their taxonomy, attributes, restrictions, and relationships related to a given domain. Ontologies are normally described with formal ontology languages such as OWL Web Ontology Language (standardised by the *W3C* [55]) and have gained importance in describing and modelling context cantered around a given domain.

The context model needed in this work is chosen so that it satisfies certain requirements. The context model has to be sufficient to describe network context in a first instance and in a standardised way. It also has to be light weight since the resulting format of the data has to be exchanged in a distributed environment with high dynamics. Possible candidate context models are next reviewed.

3.1.4 Context Models for 4G Networks

Making a mobility management system context-aware aims at enhancing the management decision process in a way to enable intelligent selection of wireless access networks. The decision process should be optimised in a way that best suits the user activity and

their situation. In this work (partly discussed in [111]), entity context is adopted to represent mobile network access elements such as access points and mobile users. Context relates to entities as defined by Dey and Adowd [24; 62] in Definition 3.1.1 or by Schmidt [165] in Definition 3.1.2. The main goal of using context information is to allow a better programmability of network connectivity at the wireless edge. In other words, users should access those networks that best suit their situation, environment, and needs.

Prehofer et al. [146; 147] have been the first to propose context-aware mobility management, as such. And in order to cope with the dynamic nature of the involved context, active networking is suggested as the way to enhance the adaptability of network elements.

Huong and Mitsuji argue that context-aware network management aims at enabling more intelligence in service provision and network management, while achieving an enriched information of users and providers [93]. In terms of context information models for supporting context-aware mobility, Huong and Mitsuji review some alternatives found in the literature. First, the early proposals by the IETF for context transfer protocol (RFC 4067) has to be mentioned. This early work suggested to transfer security, policy, QoS, header compression, and AAA information during inter-domain handover between routers and the mobile node. Second,the *resource description framework schema (RDF-S)*, could be used to declare vocabularies commonly used by a given community, and therefore, influence the definition of classes and their properties. Furthermore, the two authors also look at the use of profile-based context CC/PP [105] (used by Prehofer et al. [147] as well), which defines a two-level hierarchy of components and attribute/value pairs for describing a profile of a device or that of user preferences [93]. Another alternative is based on general user profile, which is a profile based representation of context information, defined by the 3GPP as a collection of user related data and is serialised in an XML schema.

Huong and Mitsuji [93] propose to extend RDF-S information models with using Web Ontology Language (OWL). An example of the class structures defined for the domain of networking in 4G systems, is shown in Figure 3.1. It is, however, left open how the discovery,collection, and use of the context information could be achieved in a scalable way.

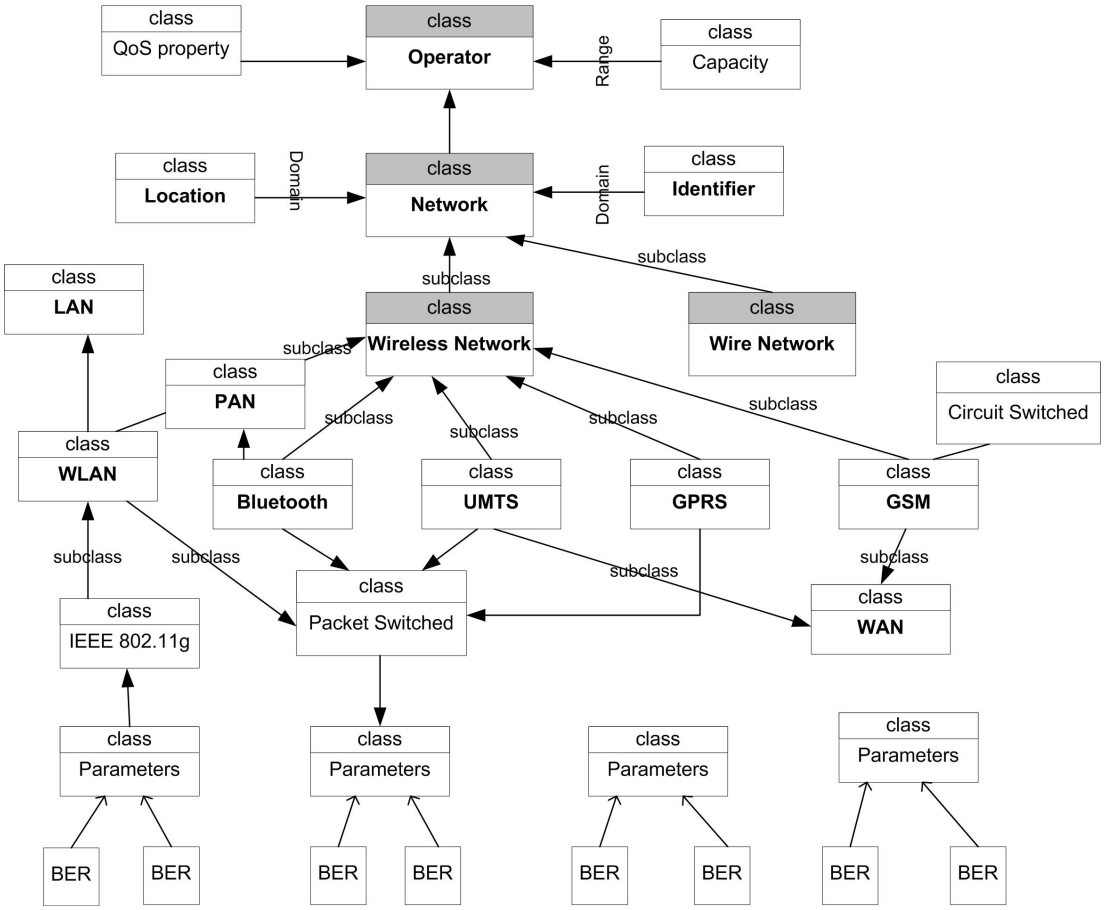

Figure 3.1: A Network Context Model - Ontology-based context model (Source[93])

The advantages of ontological management is that it creates a common, semantically rich, vocabulary for the telecom industry, which will be essential for the interoperation between providers. The semantics should offer the user the possibility to search for services among providers.

Moreira et al. [132] also mention the difficulty of pushing the telecom sector to adopt the new semantic architecture. They believe that this is going to be more driven by the content providers rather than the network providers. Other work in that direction include the CONTEXT EU- IST project and the work proposed by Serrano et al. [168].

In this work, the focus is made on the atomisation of access networks, which are reduced to entities whose context requires a descriptive language. As a result, a clear interface should be defined to describe access to each network domain. Context-aware mobility means to take advantage of the atomisation of networks into separate domains (or network entities), which are easily discovered by the mobile user. The user should then be able to adapt its connectivity service by selecting among the discovered networks entities .

The proposed architecture in this work (partly discussed in [111] [89] [91]) allows to separate user context tracking from network context management. Mobility management in context rich environments can then be defined as follows:

Definition 3.1.3 (Context-Aware Mobility) *Adapting connectivity to the wireless network whose context best matches the user context.*

Adapting user connectivity relies on the machine-readability of context. Followed by a selection process between several network contexts, a handover decision should be optimised better than just relying on pure link or physical layer mechanisms. The implication of the handover decision depends on the ability of the context reasoning or decision algorithm to interact with the lower layers through application layer triggers. The latter triggers originate from the context processing application, which attempts to match user needs and capabilities (i.e., user contexts) with the right network available in the situation of the user. The programmability of the handover process is, though, not new and originates from early proposals for programmable networks.

3.2 Programmable Handover and Active Networking

A programmable handover can be identified as part of the research efforts to enable programmable mobile networks [110], which represent a specialisation of programmable networks. Although the general term of programmable networks often refers to a large area of research, it could be defined as enabling software-configurable networking platforms. This can be compared with mainframe computers, which, in the 1970's, used to run only proprietary software supplied by their manufacturers. Networking hardware elements still operates using this approach [60]. Programmability of networking hardware suggests the possibility to adapt the hardware configuration through a clear interface. In fact, the digitalisation of communication components thanks to the advances in digital signal processing, and the use of field programmable gate arrays (FPGA), makes hardware more flexible and reconfigurable. Similarly, the ability to reconfigure the radio hardware through independent software elements also suggests that it is possible for application-level processes (like context-aware network selection modules) to interact with lower layers (as low as the physical layer, but especially the MAC, and network layers) to allow reconfiguration of wireless connectivity.

Whether from the mobile device view or that of wireless routers, reconfiguration of networking functionality also known as active networking has been pursued by several initiatives in the past. In the late nineties, the active networking research community has made several attempts in order to support programmable handovers [32; 47; 109] by using active networking agents that encapsulate handover algorithms sent to the mobile node. The latter algorithms can adapt the handover strategy to the QoS requirements which in turn are defined by the user activity.

As already explained, in order to capture user behaviour or needs, this type of information is gathered as the context of the mobile user. The situation and wireless networking environment surrounding the user can also form part of the user context according to the approach taken by the DoCoMo research group [145; 146; 147; 149; 189]. The context related to a given user combines different types of information about the situation and preferences of the user centred around three context sources, *(i) user-profile repository* (delivering user preferences), *(ii) location information* (collected dynamically), and *(iii) QoS information* (describing current application QoS parameters). These elements (illustrated in Figure 3.2), which could easily be mapped to cellular

network components, use a centralised context collection point inside the network that sends updates to handover decision point.

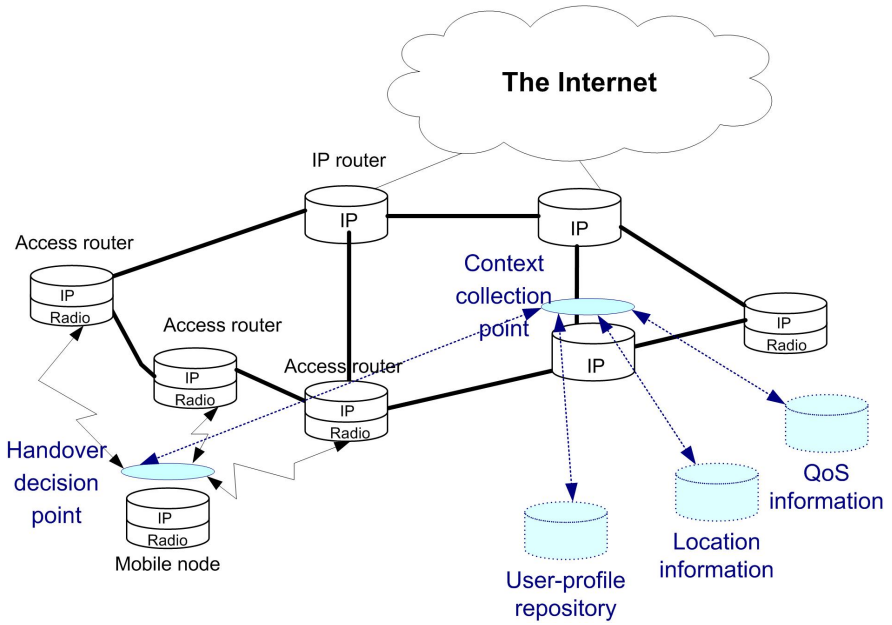

Figure 3.2: Architecture for context-aware handover - (source: [147])

To deal with context-exchange, two strategies are suggested. The first one defines a context format (e.g., XML dialect such as CC/PP [105]), to collect the user-specific context information and delivers it to the mobile node, where a handover decision application runs and uses this information to select future wireless cells. The second, method uses active networking (programmable network elements). A mobile node is made capable of receiving an active agent which is prepared in the context collection point (shown in Figure 3.2). According to [147] the following steps then take place:

1. The context collection point adds a handover algorithm and the collected context data needed at the handover decision point. The agent is sent to the decision point each time the user enters an new cell or change in the user profile occurs.

2. The agent is downloaded by the decision point.

3. The agent is invoked at the handover time. It takes as input dynamic terminal context including reachable access points, and application requests and sessions. It then provides as output a handover decision.

In fact, an implementation of active elements in both the node and network use the PromethOS active node architecture [101]. The work carried out by the DoCoMo research group [145; 146; 147; 149; 189] is targeted at an operator-centric implementation of context-awareness, where the mobile users are still managed tightly within the boundaries of the network domain and its available resources. The above approach can be categorised under the tight integration of wireless access networks in an intelligent network architecture, where the network tracks closely the creation and the running of the applications. Context-awareness is used to better deliver the service guaranties adapted to the user context while allowing some load balancing by better assignment of user traffic to the right wireless attachment technology.

The agent-based architecture removes the need to have a common protocol or a common description language to collect and describe context. The concern of scalability of such an approach remains, since context is collected not only from three architectural components, but rather from scattered information belonging to different domains. The problem of context composition is not addressed. It is though not specified how the discovery of the context information can occur in inter-domain handover management.

Furthermore, the DoCoMo approach relies on exchanging the handover algorithm, which is encapsulated in the active agent that is then uploaded to the mobile node. This approach suggests that handover algorithms have a great impact on the selection process, and therefore, updating the algorithm might be necessary. In fact, decision making algorithms are a field of research of its own. Next, the main categories of handover decision algorithms are reviewed.

3.2.1 Context-Aware Handover Algorithms in 4G-Systems

With the availability of context information, decision making or reasoning can take place. In ontology based context information, reasoning plays a major role already. The selection process of relevant context information could already limit the amount of involved context data. Often only a small portion of the collected context is actually useful in the decision process. Context reasoning is not just limited to mobility research. With the help of ontologies, some reasoning can take place, while collecting the context information and selecting those elements of context, which are useful. Once this information is made available, a context-aware handover algorithm needs to be used during the selection or triggering of a handover. Similar to classical handover algorithms, some

algorithms are better suited for certain network architectures and user behaviour than others. Based on the work carried out by Zhu and McNair [200] and more recently Kassar et al. [100], three classes of handover algorithms can be identified.

Fuzzy logic Fuzzy logic based algorithms offer a powerful way to capture both the fuzziness of the information fed into the system and that of the rules, which are taken in the decision process. Fuzzy logic can model well fuzzy states which are hard to measure or quantify. A well known example is that of coldness, which means different things depending on the described object (the weather might be felt as cold temperatures are below 10 degrees in a city like Athens, whereas in Moscow, weather coldness refers to much lower temperature). Therefore, the coldness can be said to be hard to quantify even if temperature is easily quantifiable. Additional rules are added to define what coldness means depending on the measurable air temperature. Similarly, when evaluating context information, fuzzy logic can associate rules (in the form of "if...then" statements), which can help quantify context relative the entity which is compared (in this case network context). As an example, Hou and O'Brian [87], measure network context in terms of data loss. Packet loss, however, is interpreted differently depending on the application running. Two cases are identified (through "if...then" statements). Either to start a handover triggered by loss, or not to start it. In the case of bulky applications (high data rate, like video streaming), a handover procedure results in an interruption time which leads to considerable packet drop (e.g., due to buffer overflow during handover procedures), which leads to the decision "a handover procedure should be avoided when using bulky applications, while tolerate packet loss due to short term deterioration of link quality". In the case of less bulky real-time applications like voice, the application does not tolerate short term loss, and therefore a handover procedure should take place as soon as this happens (i.e., look for a better cell as soon as loss occurs). The fuzzy logic, based algorithm as used by Hou and O'Brian [87] allows to choose between the two handover disciplines based on the fuzzy rule that applications are either bulky or not. Similarly, Huong and Mitsuji [93] also used a fuzzy logic based decision algorithm, which provides a powerful method to limit unnecessary handovers, which

besides negatively affecting bulky applications, leads to the ping-pong effect (i.e., handover back and forth).

Policy based handover According to Fan et al. [73] policies are simple condition action pairs, where certain actions are taken when the conditions of that policy are met. The conditions are applied on various context information collected from different parts of the network such as load on different layers, resource utilisations, or from the end-user equipment. Conditions are composed of arithmetic or logical statements on context information that is available at the decision making node. Multiple conditions can be combined, in a manner similar to the logical `"AND"` operation. The actions are mobility management related events that can be triggered on the network node to which the policy is associated. Multiple policies are combined in a logical `"OR"` manner, so that all the possible combinations of the context space spanned by different values of the context information is covered [73].

Earlier work defining handover policies include policy based Mobile IP handoff decision (POLIMAND) which uses link layer parameters for mobile IP handoff decisions [33]. Helen J. Wang [188] first showed policy-enabled handovers. Vidales et al. proposed PROTON [187], focused on the complexity of the policy based algorithm to support more dynamic data and then the handover procedure while still offering a light weight solution. The key motivation behind PROTON is (1) how a flexible policy-based approach is suitable for 4G scenarios, and (2) how to incorporate richer context into policies and still maintain a light weight solution appropriate for mobile devices [187].

The known algorithms gather policy based handover which could be easily integrated in in UMTS architecture through two conceptual architectural elements *policy decision point (PDP)* and *policy enforcement point (PEP)*. Policies can be defined on known cost functions, which calculate the benefit to handover to a particular network. Systems such as the MUSE-VDA algorithm explained in [200] concentrate on QoS performance and utility offered by each network with detailed performance analysis of handover process (e.g., blocking probability and average satisfied users).

Analytic hierarchy process Multiple criteria optimisation algorithms are borrowed from operational research and they apply an *analytic hierarchy process (AHP)* [159]. This algorithm offers a powerful way to classify user objectives which have to be targeted by the optimisation process. The objectives are first defined and then weighted relatively to each other. Some objectives are rated as more important than others. In [37], vertical handover is seen as the process of selecting the cell that suits several needs (mostly centred around QoS) of the user among several alternatives. Each alternative cell is evaluated according to several objectives like *(i)* minimising interruption time, *(ii)* maximising bandwidth, *(iii)* longest connectivity possible, *(iv)* minimising price of communication, *(v)* minimising jitter, etc. The latter objectives are weighted in advance according to their importance and relative to each other. This process can be quite subjective, but the result is an objective matrix whose size depends on the number of competing objectives modelled. The score each prospective cell achieves is combined in a utility function, which is used to select the best handover alternative between overlapping cells or next in line cells. Ahmed et al. [27] apply the same methodology to a GPRS - WLAN scenario and study the execution time of the vertical handover procedure which turns out to be less than 200msec.

The handover decision process for 4G should use multiple criteria for the selection process. This problem is a multi-dimensional optimisation process, with fuzzy information components and decision rules. In this work a focus is made on adding location prediction to the decision process in order to have proactive selection of wireless networks rather than a reactive one. This approach is explained in Chapter 5.

3.2.2 Context-Aware Handover Optimisation

Context-aware handover has been approached on several aspects including:

1. Ontology based context models in mobile systems.
2. Context retrieval and tracking across heterogeneous networks.
3. Decision algorithms for context-aware handover.
4. Performance improvement of context-triggered handovers.

Context Type	Context information on mobile device	Context information on network side
Static	User settings & profiles Application settings	User profile and history Network location Network capabilities and services Charging models
Static w.r.t current cell	Reachable access points	Potential next access points
Dynamic	Application requests Device status (battery, interface status, etc.)	Location information and location prediction Network status Network load

Table 3.1: Context information classification, Case study [145]

In this work user context refers to the set of descriptive attributes that identify the capabilities of the user's terminal to be self-aware of its pervasive capabilities or application needs. Combined with a set of user policies, preferences, and personalisation methods, some rules results from the combination of the MN node's capabilities and the user's preferences and behaviour. Context description of the user then attempts to find the right network connectivity that suits its rule set.

According to Definition 3.1.3, context-aware mobility management is about context matching between the mobile user context and its surrounding network context.

Figure 3.3 demonstrates the scalability and complexity problems faced by context-aware mobility management that supports inter-domain mobility. Whether a network controlled handover or mobile assisted handover is used, the context-aware part relies on a distributed system to retrieve context information from separate management domains. The retrieval process has also to be adapted to the dynamic user context. User movement tracking capabilities is an example of user dynamic context, which can influence most the retrieval process. Movement can be detected in several ways, either through positioning technology like *global positioning system (GPS)*, UMTS-positioning [197], or through interaction with location servers like Placelab [10], cellular network cell ID, and mobile IP's care of address. The accuracy the positioning technique influences the way network context is retrieved.

I approach the problem of context discovery and retrieval as a spatio-temporal query problem in a mobile environment. The query process depends on the movement of the user and the ability to predict the movement direction or movement pattern out of the cartographic information available to the user, such as street networks or possible future positions. The context-aware mobility system supports network context discovery along

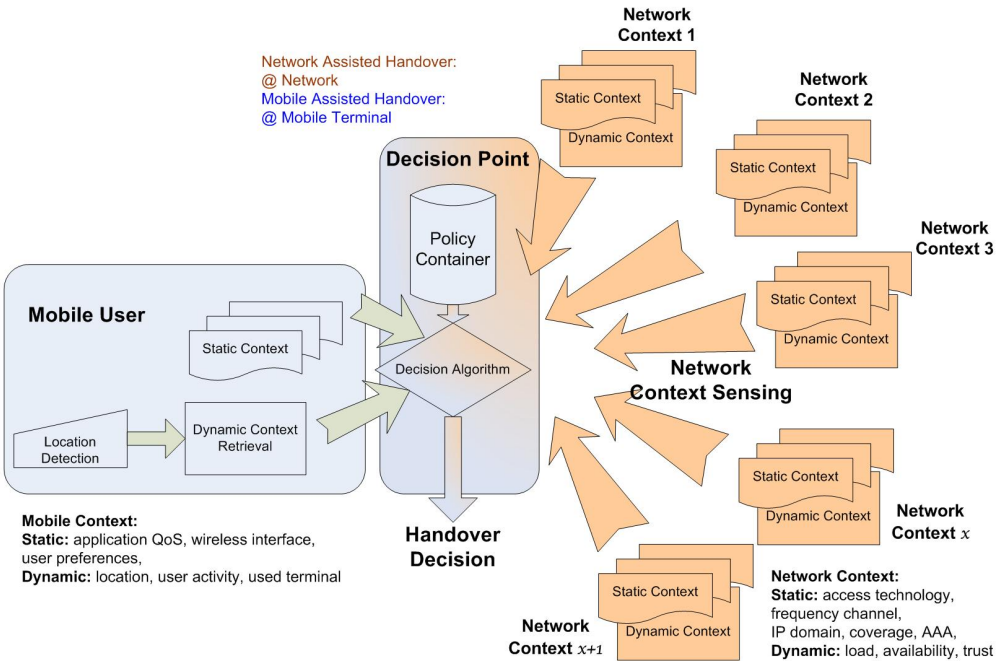

Figure 3.3: Context-aware vertical handover - Left to right: *(1)* mobile context collection and sensing, *(2)* decision point (either in network or in mobile terminal) *(3)* Network context sensing per user

the predicted movement path. Based on this assumption, network discovery could be done in a predictive way while relying on a context management system responsible of tracking and coordinating network context across heterogeneous wireless domains. The context management system could be seen as a distributed spatio-temporal database. Next, the type of queries in mobile environments are looked at.

3.2.3 Dynamic Querying in Mobile Environments

There are several types of queries which have to be supported by the context management system. Grine et al. [79] analyse the requirements of adaptive query processing in mobile environments. Two classes of mobile queries are relevant to this work:

Spatio-temporal queries when considering movement, the query results could depend on the query's spacial properties. As an example the query bound might progress in time due to the movement of the user, so the query bound is said to depend on the location of the user and the change of location with time. Normally these queries are either interested in trajectories describing a time history of the object movement, or on the current position of the moving object and possibly its future position.

Continuous Queries are types of queries that allow the user to receive new results whenever they become available. As an example being informed *continuously* by some publish/subscribe database of gas stations as soon as these are 10 kilometres away from the current position of a driver.

In terms of queries, there are three entities that can be distinguished: *(i) the mobile node* that initiates the query, *(ii) the query broking node*, and *(iii) the containers of network context information*, which respond to the query. With spatio-temporal and continuous queries, client mobility tracking is important since it means that the query boundary and query size are linked to both location but also to the change of location with time.

3.2.4 Context Composition in Heterogeneous Systems

The context management system has to cope with continuous and spatio-temporal query problem, which are used to support mobile users. Another system requirement is the

ability to compose between context sensing entities to support this type of queries. Sensing and retrieving network context from scattered domains could be redefined as a context-rich location-based service (LBS).

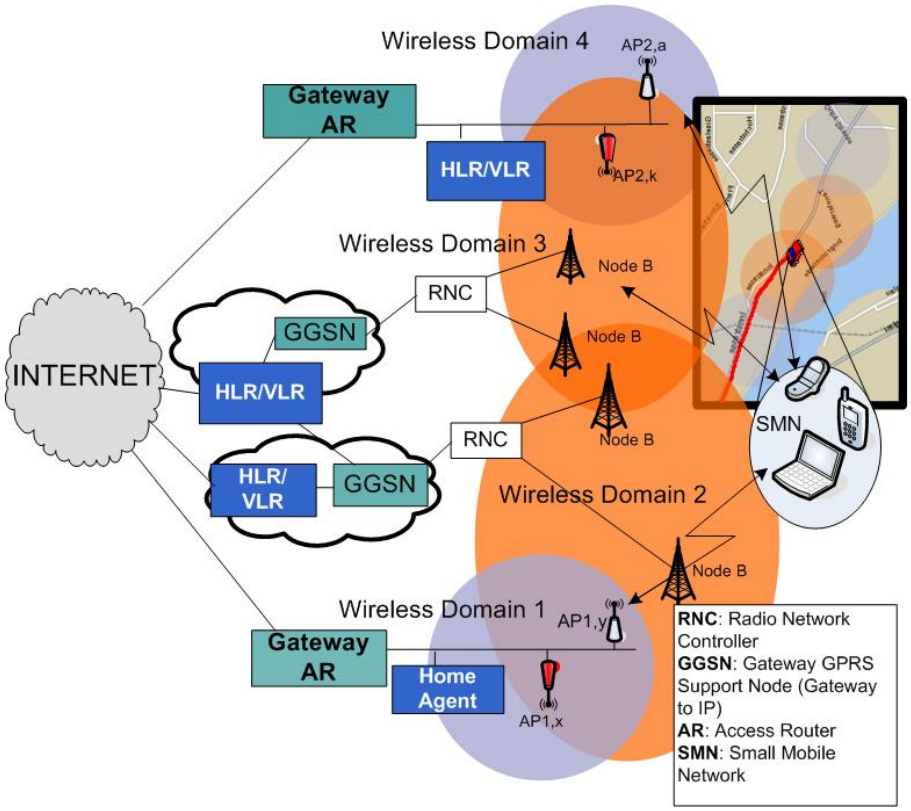

Figure 3.4: Domain aggregation of wireless networks - Semantic boundary between different wireless network domains

The discovery of network context has to occur across heterogeneous systems. Therefore, a middleware bridging between the different networks is needed. The middleware offers a common interface to represent the heterogeneous wireless domains and their capabilities as meta data. The network context has to be discovered and retrieved while adapting the two processes to the user context. The result of the discovery effort should be all relevant network context centred on wireless autonomous domains (shown in Figure 3.4. The characteristics of a domain can include:

- Each wireless autonomous domain should reflect geographic proximity of wireless attachment points (i.e., a continued island of wireless coverage is guaranteed via

a group of access elements).

- Given a set of link technology used within each domain, some mobility management capabilities are required to describe the connectivity and the intra domain mobility support offered. An example would be to describe a UMTS domain as offering low bandwidth with little variability and minimal interruption, loss, and supporting high user velocity. In comparison, a WMN based on IEEE 802.11 might offer higher bandwidth, while incuring a much lower financial cost per data flow, but only supporting a lower user movement velocity.

- Some notion of load and or availability of wireless resources needs to be derived for each wireless domain. Extracting context information describing load is hard, since the notion can vary from one wireless domain to the other. As an example, in cellular networks, load can be measured per cell as the current number of admitted users. However, in WMNs this is less obvious. A recent proposal attempts to aggregate traffic along well provisioned paths until reaching the wireless gateway connecting to the Internet. The traffic engineering approach is given in [43; 68], which allocates mobile users to given parts of a WMN depending on the resources available at the Internet gateway connecting the mobile mesh client to the Internet. The above approach is an example of how aggregating IP-level flows, with the help of traffic engineering methods, could be a good indicator of load per autonomous system.

- In heterogeneous wireless networks, each domain should include only those base stations implementing the same link layer protocols.

- Separating domains, including within a single link technology, apply the following heuristic: handover cost between base stations within a given domain should be considerably smaller than that between two domains. Handover cost can be represented by several metrics including the handover delay, route adaptation effort, re-addressing and reconfiguration effort, etc.

A context management that allows mobile users to receive updates on upcoming heterogeneous wireless domains, can be built as a centralised or a decentralised discovery and retrieval system [1].

[1] This refers to the backend server architecture used for the network context management, which

3.3 Centralised vs. Decentralised Service Discovery

In order to understand the need for decentralisation, the needs of a context-management systems are compared with other distributed systems found in the Internet. The level of decentralisation of control or composition between the distributed resources/services depends on the different examples. In other words, not all distributed systems use a decentralised composition or control mechanism. Many distributed systems are supported by centralised mechanisms.

3.3.1 Centralised Approach

Even though the context information might originate from distributed sensing elements, the mediation or broking effort could be done in a centralised element. Instead of querying each domain separately, the central mediation or index system is involved. The mobility management system only needs to register to the central registry. The data repository can be constructed in several ways. Either *(i)* the data providers register with a central mediator or broker, e.g., Elvin [11] using a publish/subscribe content-based notification system, and Napster [12] as a peer-to-peer file sharing with a central mediation point; or *(ii)* active mediators and crawlers that actively search for matching content, e.g., search engines such as google or yahoo especially crawl the web periodically to find content. The users of such centralised systems submit their query to the mediation point, which resolves the query by accessing information containers or database. An example of a centralised approach would be that of the FON network [13], which is a community-based WiFi sharing network. In order to join the FON network, home users install a FON access point, whose status and access rights are managed by the FON network. The availability of each single access point is tracked and added to a centralised map service, built as a Google maps mashup. The user of the FON network can access the FON maps service to locate the nearest FON access point.

There are several problems with centralised solutions that are mostly related to scalability. These systems might face problems especially in the case of having dynamic updates to the content, or when increasing the number of queries. Constructing data grids and large computation clusters with well provisioned network links, help to limit the problem.

should support continuous and spatio-temporal queries on behalf of mobile users

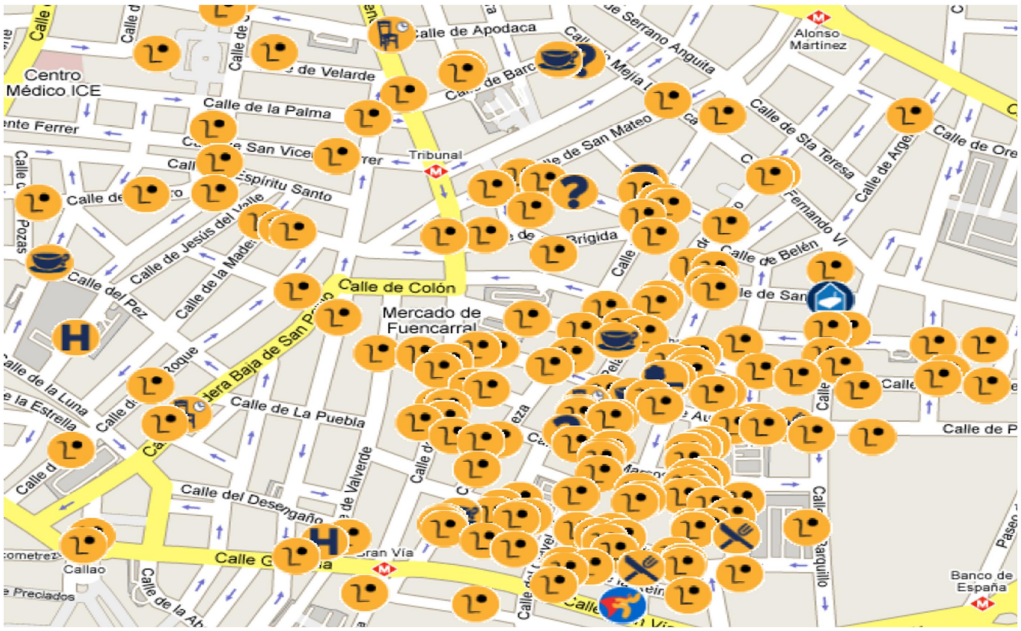

Figure 3.5: FON maps broker service - Google maps based service illustrating an area in downtown Madrid (Spain), source http:maps.fon.com

The arrival rates of queries in a mobile system is also a lot higher than that in the fixed Internet. A centralised database has to satisfy spatio-temporal queries which are supposed in the worst case be generated by each user separately with an arrival rate from each user higher than that of DNS queries. In the best case, the user has a navigation device which defines a full movement path from start to finish. The query is then started at once. In the worst case, the user is only aware of her/his current position and possibly the direction of movement, which might suddenly change at each street junction. The user is supposed to query the data base system each time a discrete change in movement is detected.

A centralised system also forms a single point of failure as well as a security risk. If a mediation point is down or is attacked, the whole system is made unavailable. Furthermore, broking secure content makes it even more vulnerable to intrusion attacks, where an attacker having infiltrated a centralised system can find out information about secure information supplied by several providers at a time.

It is also worth remembering that the deployment cost of a centralised service is very high requiring corporate support [76], which is the opposite case for peer-to-peer

systems where all participants carry out both storage and mediation of the content incurring limited additional investment.

3.3.2 Decentralised Alternatives

The goal of the context-management system is to register context originating from autonomous systems and allowing asynchronous updates of the data content. On the other hand, mediation has to cater for the needs of the users, their location, and their movement. The mediation effort can also be carried out in a distributed and decentralised manner. According to [76], decentralised mediation systems found in the Internet can be categorised as: *(i)* hierarchical tree based systems, *(ii)* unstructured overlay networks, and *(iii)* structured overlays.

3.3.2.1 Hierarchical Tree-based Systems

A well known example of a hierarchical mediation structure is DNS (domain name service). Hosts belonging to the same domain are aggregated together, and a node can then be assigned to store $(name, IP)$ tuples. DNS builds a tree structure between DNS servers world wide. Users of DNS query the system in order to look up the IP address corresponding to a given DNS name, therefore, if given the possible name space, queries in that space are randomly distributed and are not clustered. In other words, DNS queries can involve DNS servers which are often far away from the DNS domain originating the query and might be distributed worldwide. A local DNS server maintains a portion of the possible DNS names, and so does the DNS sever one level of hierarchy above. Theoretically, the root DNS server should be able to translate any DNS name or at least knows the branch of the DNS tree responsible of the name.

The main advantage of the tree topology is that system-wide content broadcasting and query flooding are avoided. Hosts in the tree structure store only a portion of the content managed. The content is not broadcasted to different parts of the network but rather concentrated in subtrees using some aggregation criteria of the data stored. The DNS, in this sense, stores names of Internet hosts with their IP address. The next level of hierarchy gathers all root nodes belonging to the same organisation or within the same DNS branch (which could be organised according to countries, DNS name structure). In DNS, the tree structure is useful to start a query anywhere in the tree (i.e., at any given branch) to reach any other branch. The requesting node has to send

a query to its known root DNS server, and the latter forwards the query to its root DNS server until reaching the right level of the hierarchy common to both requested DNS name and requesting node. Since the information in that tree is rather static, there is less need for updates of the tree structure. Some caching at local root DNS of past query would reduce the number of required queries higher up the hierarchy.

The problem with such hierarchical structures is that the nodes higher up the hierarchy have to deal with a larger load than those at a lower hierarchical level. Also the tree is not robust against node failure, since once a node fails, this will lead to a tree split. Such systems cannot deal well with dynamic content or restructuring of the content assigned to the leaves of the system, since caching becomes harder and more updates are needed.

3.3.2.2 Structured and Unstructured Overlays

Overlays are organised into general graphs which allow routing from any given node to any other node connected by the graph. They build a network where routing decision are taken by each overlay node locally. The amount of routing information stored by each node and the number of hops depends on whether structured of unstructured protocols are used. Each node in the overlay network usually keeps track of those items (or content) locally stored. It can also include a limited amount of routing information to both other overlay nodes and their content. Overlays can be seen as load-balanced storage systems, which are robust to node failures and to updates to dynamic content. However, this comes at a cost which is the querying effort, which usually involves all nodes storing the queried content and the intermediate nodes involved in the overlay routing. Updates to content might require broadcast or at least directed broadcast to update routing information at other overlay nodes.

3.4 A Case for Overlay-Based Context Management

A wireless network database for heterogeneous systems is fed by decentralised sources of information, but the mediation effort can be concentrated locally, and is geographically bounded to the location of the user and that of his/her predicted movement path. Given the location of the user, the database system provides information about local wireless networks.

It would be sufficient to centrally concentrate all mediation information in a local server. A mobile node then has to direct the query to the central index which responds with the a list of corresponding local servers that match the query. However, such a system faces a serious scalability problem given the nature of the queries in mobile environments and the number of hypothetical users of the system. In addition to that, the number of wireless domains which have to periodically send updates (such as keep alive messages) would already put the central server under high strain.

The way to reduce the load on each server is to limit the scope of each database server to a given geographic subspace which gathers local information on all resources located within its assigned zone (or subspace). This could be compared with each leaf DNS server assigned to a given autonomous IP organisation, (i.e., partition of concern) and can deal with local queries and then decides whether other root servers need to be involved. The problem with the targeted scenario is that user mobility imposes both spatio-temporal and continuous queries which are linked to the movement and position of the user. And furthermore, the handover to a given wireless network also requires the query session to move to a new server constantly. Session mobility is therefore a complex functionality which adds to the complexity of dealing with a single user in this spatio-temporal decentralised database system.

I argue in the next part of this chapter that the use of overlays can prove more efficient in dealing with spatio-temporal queries. The need to flood servers dealing with neighbouring geographic zones is removed. Instead, the query is guided through the structure of the overlay. The overlay's role is also to separate the scope of user mobility and its effect on the query session mobility. In other words, if the user moves out of the scope of a database server, the server is not aware of that. The distributed servers are however still structured to effectively manage the dynamic content (in this case network context). Since these servers might originate from several domains, their geographic scope can overlap as that in Figure 3.6. Connecting these overlapping servers would scale better when using an overlay which builds a network graph that can support range queries for mobile users.

An important aspect of improving data discovery to an overlay system is how to structure the overlay so that the number of nodes involved in the querying process is as close as possible to the number of nodes storing the data matching the query [166]. For this, overlays have to index data in a way so to reduce the amount of traffic needed for

data discovery (querying) and routing. Data similarity and data modelling play a major role in constructing the right indexing and partition of data space and its mapping to the overlay indexing system. Such overlays which map data semantics to their structure are called *semantic overlays*.

3.4.1 Semantic Overlays

Crespo et al. [54] define a semantic overlay as *"a flexible network organization that improves query performance while maintaining a high degree of node autonomy"* . A semantic overlay aims at improving the querying process in decentralised systems while constructing overlay structures based on the semantics of the managed resources. The type of queries that are supported by a semantic overlay are constrained by the following system requirements:

- Look up heterogeneous wireless cells among different operators.
- Look up only those wireless cells near the movement path of the user.
- Look up those wireless cells that the end device is capable to support.
- Limit the query path in the network for timeliness considerations (i.e., a result of a query is only relevant if it is sent back before the user has moved out of that cell). A wireless resource status could change which means that the query result is more likely to be invalid if the query occurred too long before the position has changed.

Selecting the right semantic attribute to cluster the nodes is also important. Assuming a entity context model that describes context information related to each network entity. A network entity is context description of a wireless network domain. Context information is gathered along various context dimensions defined as a list of attributes. Some of these attributes can be defined by all networks such as location, operator, link technology, etc. Other attributes could extend the context information along dimensions which are specific to a given network or technology.

Assuming that each entity context is a multi-attribute object o_x, described through its attributes' set $\mathbb{A} = \{a_1, a_2, a_3, ..., a_i, ..., a_{max}\}$, where a_i is the associated i^{th} context dimension. Each context dimension is represented as a type of attribute, which could

be associated with a set of possible values $\bar{a_i} \in \mathbb{V}_i = \{v_1, v_2, v_3, ..., v_{max}\}$. Each object o_x is described by an attribute value for each context dimension i. Each attribute value $\bar{a_i}$ is mapped to an integer set $S_{a_i} \subset \mathbb{N}$, whose size represents the number of possible values associated with a given attribute a_i.

An example would be to count all possible wireless link technologies (assuming that there are 20 known link technologies), the attribute values are given by the variable $\bar{a_i}$ for link technology $\overline{a_{link}} \in \{GPRS, EDGE, UMTS, ..., WiMAX\}$. It is sufficient to define $S_{a_{link}} = \{0, 1, 2, 3, ..., 19\}$ whose elements k_{link} are mapped to elements of the set $\mathbb{V}_{link}$ by the mapping f_{link}, where $f_{link} : \overline{a_{link}} \rightarrow k_{link},\ where\ f_{link}(\overline{a_{link}}) = k_{link}$ and $\overline{a_{link}} \in \mathbb{V}_{link}$ and $k_{link} \in S_{link}$.

The mapping f_{link} maps elements of the set $S_{a_{link}}$ to elements of attribute values $\mathbb{V}_{link} = \{GPRS, EDGE, UMTS, ..., WiMAX\}$. This number represents the key k_{link} associated with the value $\overline{a_{link}}$ through the indexing function f_{link}. The order of elements in the set S_{a_i} is important and should reflect semantic neighbourhood between different objects along a the chosen attribute. For example objects whose *link technology* attribute is GPRS is semantically closer to UMTS. WiMAX is also semantically closer to WiFi than to GPRS. This effect is reflected by the key in $S_{a_{link}}$ corresponding to the above examples.

More formally, since $S_{a_i} \subset \mathbb{N}$, the attribute value mapped to a key $k_i \in S_{a_i}$ is a natural number. Any objects whose attribute in the dimension i assigned the value a_i is mapped by $k-1$ or $k+1$ are said to be semantic direct neighbours to all objects whose i^{th} attribute values $\overline{a_{link}}$ are mapped to k, by the mapping f_i. Semantic neighbourhood can be formally described for any two objects o_x and $o_{x'}$ along the i^{th} context dimension. For the objects o_x and $o_{x'}$, their corresponding i^{th} attribute values $\bar{a_i}$ and $\bar{a_i'}$ are mapped to keys (k, k'). The two objects are said to be semantically neighbours along the i^{th} dimension iff $((k, k') \in S_{a_i} \times S_{a_i} \wedge 0 \leq |k - k'| \leq \sigma)$, where σ is a small integer. For $\sigma = 0$ the two objects o_x and $o_{x'}$ have an identical attribute value. For $\sigma = 1$ the two objects are said to be direct neighbours along the i^{th} context dimension.

This model is then used to cluster network context into a finite number of subsets, based on a selected attribute a_i whose values $\bar{a_i}$ are mapped to the set S_{a_i}. The latter set can be split into equally sized subsets. Each subset could be assigned to a semantic overlay node. For network context, the chosen attribute has to satisfy several requirements:

1. The chosen attribute is common to all possible objects, but its value $\bar{a_i}$ is mapped per object o_x to the space S_{a_i} (e.g., all networks use some link technology which is described by an attribute a_{link}).

2. The common attribute might be described with the elements of a space S_{a_i} that could itself be split into subspaces $S^0_{a_i} \bigcup S^1_{a_i} \bigcup S^2_{a_i} \bigcup, ..., \bigcup S^n_{a_i} = S_{a_i}$, where n the maximum number of key subsets of S_{a_i}.

3. Each subset is fully independent of other subspaces ($S^0_{a_i} \bigcap S^1_{a_i} \bigcap, ..., \bigcap S^n_{a_i} = \varnothing$)

4. Each subset $S^j_{a_i}$ could be assigned to a given overlay node through some indexing scheme. The latter overlay node is responsible of all objects whose i^{th} attribute value is mapped to the the subset $S^j_{a_i}$, managed by the overlay node.

One alternative to structure the semantic overlay is to select one suitable context dimension i, and assigning subsets $S^j_{a_i}$ to given nodes to manage. The overlay node is structured along the attribute dimension a_i, whose values $\bar{a_i}$ are mapped to keys in $S^j_{a_i}$. The indexing or mapping used to between overlay nodes and the semantic dimensions has to be tailored to the query behaviour as well the logical distance between the node inserting the object in the overlay and its assigned overlay node.

The choice of semantic attribute to structure the overlay could be for instance the wireless technology attribute, so that each object, whose link technology is the same, are clustered together. Although this would satisfy the conditions above, the number of subspaces that result is quite limited. In addition to that, clustered objects representing the same technology would not scale, since this means a mobile user moving through the city of Berlin has to contact the same overlay node as a user from New York. The next possibility would be to achieve an aggregation along the lines of operator boundaries. However, the same problem of scalability would arise when users within the same country need to query the same node representing all network contexts representing a given operator. An example of the overlay structure is illustrated in Figure 3.6

The aggregation of the multiple attribute objects is done along the *link technology* attribute would result in 20 subsets whose size is 1. These subsets are $S^0_{a_{link}}$, $S^1_{a_{link}}$, ..., $S^{19}_{a_{link}}$. A semantic overlay could be formed by assigning each subset to a node or a group of neighbouring nodes. A network entity, described as the tuple $network\_context = \{a_1, a_2, a_3, ..., a_i, ..\}$ is grouped with other network descriptions that share the value

$\overline{a_{link}}$. In the case of the link technology attribute, the number of elements of the subsets $S^0_{a_{link}}$, $S^1_{a_{link}}$, ..., $S^{19}_{a_{link}}$ is 1.

The more intuitive aggregating attribute is the location attribute, therefore those objects belonging to any operator and using any type of technology are clustered together if they are geographically close to each other or colocated together. The number of subspaces required to define the possible clusters is a matter of space modelling, which will build on the requirements or restrictions of a P2P routing protocol and the structure of the P2P network.

When compared with DNS, the network context system connects several heterogeneous systems requiring an interoperability layer that translates the description information generated by the various management systems into a single format. This meta information is then updated dynamically by the different domains. Each wireless domain is described through its dynamic meta data. The meta data is described through its geographic position which is fixed.

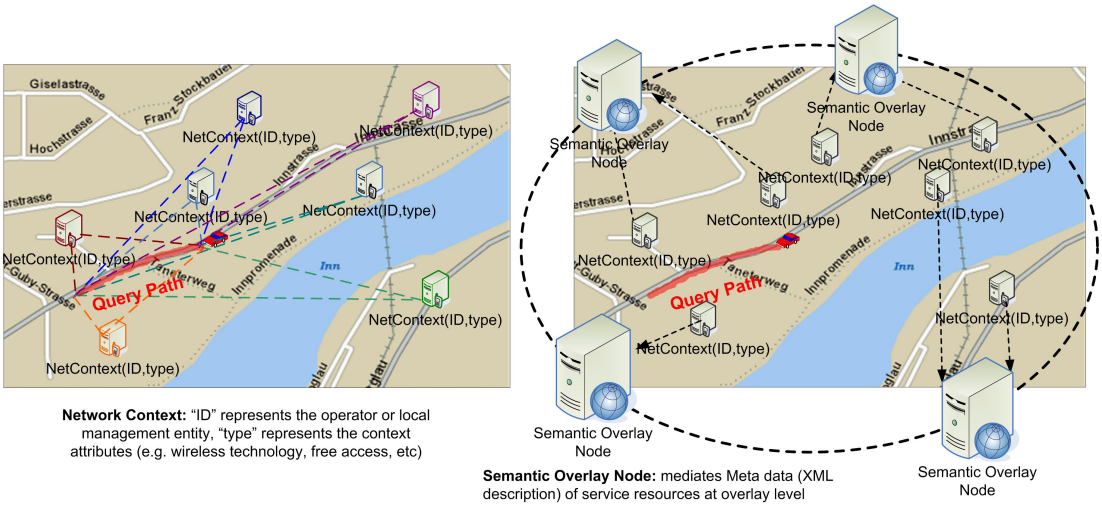

Figure 3.6: Collecting network context from heterogeneous domains - Spacial query definition

3.4.2 Context Sensing and Insertion in a Semantic Overlay

Context information is generated for a wireless domains which are tracked and described by some management nodes. These nodes take the role of transforming management

information into network context information. The generated context is an entity context with multiple attributes, where each attribute describes a given context dimensions (following the terminology introduced in Section 3.1).

The entity context description is generated (e.g., XML description). Although the format of the context description files is out of the scope of this work, therefore this can be taken as an example only. Other ontology descriptions can be thought of. The main focus of this work is in selecting the right structure for a semantic overlay capable in achieving the following goals:

1. Support data indexing along one or several context dimensions.

2. Efficient semantic partition of data among overlay nodes. This has two main aspects: *(i)* the communication cost of inserting data items, which are dynamic in nature in the overlay; *(ii)* better support for spatio-temporal and continuous queries.

3. Despite user mobility query definition and responses have to reach the user at whatever point in the network he/she currently is. This requires a mobility support at the overlay level too.

Structuring overlay networks to support semantically rich queries is not a new topic. For the sake of range querying of semantically similar content, several proposals have come out of the P2P applications.

P2P applications have evolved [150] since the days of Napster as a file sharing technique between equal entities named peers, to become one of the fastest growing Internet technologies. The key aspect of interest in P2P is the decentralised location of objects queried through a common interface that allows definition of some semantic rich queries. The structure of the P2P network and the routing of queries relies on specialised protocols. With Napster, a centralised approach was taken to store a the whole ID set used to index files in a centralised server. The other extreme, has been to distribute the index space randomly among any node that previously queried or stored part of the index set. The result is the Gnutella network. Although Gnutella might scale in terms of distribution of load and allows those files that are most popular to be indexed at a higher number of nodes, the maintenance overhead is very large. A more systematic and generic approach is based on structured P2P networks. Normally,

structure refers to the network graph which results out of connecting some overlay nodes with each other to achieve a small world topology. A well known and proven networking technique is based on distributed hash tables (DHTs) [150].

DHTs also offer an indexing functions (normally a hash function), which maps both content values and network ID to the hash space representing a finite key set of content and node ID. The hash values are distributed among overlay nodes, resulting into distributed hash tables. Several DHTs exist which tried to optimise distribution of similar data items among peers or overlay nodes.

A peer is a computational entity (which could be a virtual one) capable of mediating information about resources (in this case wireless cells and their status) stored or managed by the underlying server infrastructure (as shown in Figure 3.7). The

3.4.2.1 Managing Context through a Semantic Overlay

Adding the object to the DHT object plane is done through a node within the wireless domain capable of tracking the network context (especially those dynamic attributes such as load, number of connected users, etc). A separation is made between nodes pushing meta data and those storing nodes which belong to the mediation overlay. The managing node pushes the network context description into the DHT object plane by requesting a key or set of keys needed to index the object from the storing nodes. The meta data is pushed with a validity state lasting τ_{valid}. The storing overlay node, called *"access peer"*in Figure 3.7, has to store the content physically, storing a copy of the description for the time τ_{valid}. The DHT interface is then used to access those XML descriptions by queries generated for the mobile nodes. The resources found on a given peer are checked for the following: *(a)* validity of τ_{valid}, *(b)* further attribute match specified in the query message (e.g., preferred wireless technology, radio capabilities, etc). More query options could be possible. The access peer generates a response aggregating all corresponding XML descriptions. A single response message is generated by each peer back to the mobile node.

The DHT is a common aggregating layer in large scale and distributed database. More sophisticated query processing besides matching location attributes can further take place. The queries could include "if ... then" requests as well as attribute matches such as: the preferred wireless technologies at the mobile terminal, user credentials stating which services the user is allowed to access, etc. These types of geographic

queries assume a processing of the query request beyond a simple key matching as it is the case for file sharing applications.

3.4.2.2 DHT Alternatives for Structuring a Semantic Overlay

One of the reasons of using DHTs in P2P applications is their ability to provide a scalable system for storing large numbers of data items, while minimising the routing cost incurred to lookup any node or content. This lookup functionality is called mediation. DHTS also offer a load-balanced indexing of data items among peers. Chord [175], for instance, requires each peer node to store routing information for $O(log(N_p))$ other peers in the form of a hash table (each hash key represents the ID according the Chord key set), where N_p is the total number of participating peers. Chord is an overlay since its key space runs as an overlay on top of the Internet. For node IDs, an IP address is stored in the hash table. A Chord single hop might involve several underlaying IP hops. The mediation effort in Chord takes a maximum of $O(log(N_p))$ routing messages to find any content or node in a Chord ring, given that each node manages about the same amount of keys.

In comparison with a P2P file-sharing application, the content shared in a semantic overlay should preserve data similarity when mapping content to overlay nodes. However, in Chord for instance, a hash function such as $SHA-1$ is used. Hash functions are known to use bit transformation, where any length bit sequence can be mapped to a unique bit sequence of a fixed length m-bit. However, a single bit change of the original sequence leads to a totally different bit sequence. Therefore, when applying the $SHA-1$ to semantically near attributes, their mapped key are randomly placed in the 2^m key space. The assumption in Chord, so that a logarithmic mediation cost can be achieved is that nodes have about the same number of keys that they actually manage. This makes the hash function a good way to distribute objects to nodes in a load balanced manner.

The node responsible of the subset of the total 2^m Chord keys stores all objects whose attribute (file name, location, or any other semantic attribute) to the latter key subset. The node then stores the tuple $(key, value)$, where the value represents either the file itself or the way to get to it (e.g., IP address and port number). The application uses a $get(key)$ method, at the DHT interface, to search for the file/resource.

The random partition of object values and their corresponding keys, result in the loss of any semantic relationship that exists between the objects gathered in a node. Although all content or objects could be reached via the overlay (given their known value), the $log(N_p)$ effort has to be repeated for each object.

Besides dealing with efficient routing and mediation effort aimed at supporting a mobile semantic overlay, the responses are sent to decision making instance, which might be mobile. Supporting spatio-temporal queries for mobile agents has to be guaranteed.

3.4.3 Generic Mobility Solution: Overlay Abstraction

Based on the ROAM architecture [201], an overlay can be built to cater for mobility management. ROAM [201] is designed as replacement of mobile IP centralised home agent. However, differently to the home agent, the mobile updates its location to the whole overlay. Any logical peer can take the home agent's role as location registry without being bounded to one centralised node. Redirecting traffic to the user's location (i.e., network of attachment) relies on overlay routing. In ROAM the overlay as a whole takes care of routing and all data packets are forwarded to user's current proxy (which is an overlay node). A ROAM based overlay can be used to support user mobility in overlay networks, so that content managing peers do not have to care for user's current location. If the mobile user was assumed to be a normal peer, then its addressing tuple (key, IP_{addr}) in the overlay network would change regularly. Keeping this change hidden, means that mobility state has to be managed differently. ROAM offers a separate overlay infrastructure which deals with this issue.

In an overlay supporting spatio-temporal queries, a user query is generated with some mobility assumption. Once a response is sent back, the user might have already changed his/her position (leading to an IP update). Therefore, an overlay node acting as a proxy to the mobile node, requires two communication sessions for each user:

- A communication session for user context update, which is based on ROAM.
- A communication session to allow retrieval of network context, where query definition and response filtering needs to take place.

These functionalities are managed in a overlay node named *"Search Peer"*. Whereas access peers partition an object space among themselves along a context dimension,

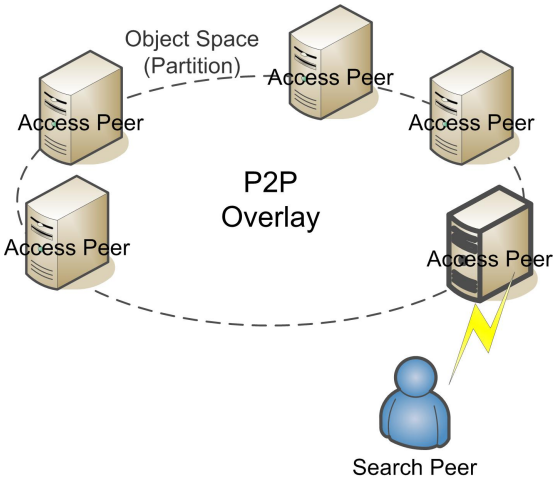

Figure 3.7: P2P solution for organising topology servers - Connecting the object space with the mobile user overlay

search peers partition user management among themselves and might represent several mobile users simultaneously. This is illustrated in Figure 3.7.

3.4.3.1 Search Peers

There are two logical overlay networks interworked with each other. The one overlay connecting the search peers and the second connecting the access peers. The ROAM protocol deals with mobility management or in other words, keeping track of user context at search peer level. The network context is managed by a structured semantic overlay optimised for retrieving network information. Search peers, although forming a different type of overlay, connect to access peers for query purposes. It represents a moving node that both identifies the context of the mobile user and his/her current node and is meant to search and retrieve "nearby" wireless network resources.

The search peer can include other functionalities such as caching query responses, and dealing with the mobility management of aggregate mobile users. Aggregation of query results could be cached in a search peer to additionally reduce the query efforts or mediation. Such issues are similar to the architecture restrictions defined the MoPi project [31; 99]. Although, the MoPi architecture has been destined to support mobile P2P file sharing applications, some similarities could be found is the way query aggregation and caching can reduce mediation efforts for mobile peers. Content sharing in P2P

file sharing has a heavy tailed distribution of requests for content. In a geographically aggregated content, users are only interested in content or information related to the location, which is common to all users located to in the same area. The dynamic nature of the content in the case of wireless network descriptions, requires periodic updates of the cache at the search peers. And for a more up-to-date view on nearby network condition, a query per user might be necessary. Although this is the assumption taken in the evaluation process of a single query or update process, aggregation or reuse of query results would also add to the scalability of the system. This is also known as sharing or federating of a context marketplace [141].

3.4.3.2 User Context Aggregation at the Search Peer

The search peer hands a session related to a given user to another search peer, when the query parameters and user context related to the urban subspace have changed substantially. In fact, this step is only optional since a mobile user can still access the same search peer during the whole time. The mobile user only needs to discover a search peer near its geographic context during bootstrapping. However, if caching is used, it is better to divide the search peers among the subspaces managed by access peers. On bootstrapping, the mobile user connects to its last known search peer which directs the new session to another search peer closer geographically to the mobile node itself. Caching strategies can then be defined to suit the dynamic nature of the content or to cater for special static queries. Ku et al. look at caching techniques for mobile LBS broadcast systems [112].

The user, despite his/her dynamic care-of-address (IP address), keeps a fixed ID in the overlay. In the case of a micro-mobility scenario the mobile node may often change its IP address, as a results of attaching to several IP domains. Adding session migration to the search peer, allows to limit the geographic distance between the mobile node and its search peer. In other words user aggregation occurs at the search peer in order to reduce the context mismatch between those cached network context and the user query needs.

3.5 Chapter Summary

This chapter investigates the way context-awareness can be applied to mobility management. The way context-aware handover techniques have evolved still left out the complexity of context collection and composition from across heterogeneous network management domains. Networking in context-aware systems is a challenging task. This chapter identifies the challenges of context sensing and context retrieval in complex systems such as networks. Context-aware networking is about offering customised networking services to users depending on their context and the network diversity surrounding the user.

Diversity is a key argument why networks need to be described as atomic entities. For wireless networks, the entities offer diverse types of networking services, which can be captured as entity context. Entity context hides away the complexity of the underlying system, whose level of detail can differ greatly (e.g., a whole UMTS network can be modelled as an entity, or a single WLAN access point might form a networking entity).

Context-aware mobility is about discovering network entities, whose context match that of the mobile user. In order to manage both user and network context, it is important to understand the nature of the context generated at network level, the way it has to be collected in a scalable manner, and the way it has to be delivered to the end user.

Network context is gathered from scattered systems and composed on demand to suit the user. This composition can be seen as a mobile location-based service and relies on a semantic overlay that organises and structures network context. The selected indexing scheme used to both structure the overlay network and to support range queries needs to fulfil the following requirements:

1. Semantic neighbourhood between objects has to be maintained. In the case of the geographic attribute, the aggregation criteria is easily describable through the notion of geographic closeness, which can be any two overlay nodes that are close to each other in terms of overlay hops should also manage objects, which are geographically close to each other.

2. The structure of the overlay has to be as close as possible to an ideal semantic overlay. A semantic overlay can be said to be ideal if for a range query, only those nodes that store the content are involved in the search and routing process. How close a selected overlay to this ideal defines an essential evaluation criteria.

3. The user mobility has to have as little effect as possible on the dynamics of the overlay and its maintenance.

In the next chapter, space modelling is used to allow allocation of network entities to geographic management subspaces. Simulation and analytic studies are carried out to identify the most important optimisation parameters of the system.

Chapter 4

Generalised Semantic Overlays for Mobile P2P Location Based Services

This chapter presents a design methodology to construct and evaluate a scalable semantic overlay, whose goal is to manage and mediate context information in mobile environments. In order to guarantee efficient mediation and routing effort, *distributed hash tables (DHTs)* are used. The DHT structure is designed in this chapter to offer a suitable mapping between the semantic structure of the managed data (i.e. network context) and the semantic overlay. Furthermore, the DHT structure is designed in order to limit the associated overhead of using overlays in a mobile environment. Aggregating context information along the location attribute has been discussed in Chapter 3. The design of a dedicated semantic overlay capable of managing the context can be generalised to other decentralised P2P *location based service (LBS)*. A peer refers to any separate management entity that offers geographically-linked information, called *geographic information servers (GISs)*. The semantic overlay takes the form of a P2P network, which can access and compose between heterogeneous and separate GIS.

Information indexing and modelling in standard GIS is first explained in order to inspire indexing in a distributed P2P LBS. The role of DHTs is also investigated before the approach taken in this thesis is explained.

In fact, a layered solution is thoroughly developed, implemented, and evaluated in this chapter. A mapping is proposed between the P2P protocol layer (and in particular

the DHT), and the managed data. The mapping is based on a separation of the 2-dimensional space, representing location, *space* from both the data plane and the overlay plane. Separately, space is modelled as a binary index using the Hilbert space filling curve, which could be described as a fractal representation of the complete 2-dimensional surface of the earth. How the space index is then mapped to nodes or how data is mapped to the space index is based on a self-organising distributed approach, explained in this chapter.

The evaluation of the designed overlay follows a numerical estimation of communication overhead. The design decisions, such as the accuracy in mapping data level information to the overlay plane, are also investigated. The resulting mapping technique is shown to fulfill the system requirements of a semantic overlay used to manage mobile P2P LBS.

In order to evaluate the influence of the mobile user behaviour on the system, a simulation is also implemented. The simulation results confirm the numerical overhead bounds, and the need for hierarchy in the DHT.

4.1 GIS Design in Mobile LBS

Mobile LBSs support service provision while taking user movement into account. The service delivered changes with the movement and location of the user. Movement tracking is usually done with the help of a tracking hardware like GPS, which tracks the location attribute of the device accessed by the user. The same attribute has to be reported to the LBS provider in order to adapt the service.

It has already been discussed in Chapter 3 that location is a suitable structuring aspect of network context. Retrieving network context at the user side (as suggested by the IEEE 802.21) is carried out on a continuous basis, while being adapted to the user dynamic behaviour. Based on the position and movement vector of the user, a query is started to search for available heterogeneous wireless cells belonging to various operators. The discovery of layout of wireless cells allows the mobile node to select a wireless cell that most suits its context. The LBS information is provided by heterogeneous management domains, and therefore the semantic overlay is used to structure the mediation effort.

The same design process used in this chapter can be generalised for other mobile LBS, which compose, in a spontaneous manner, between new semantically different or administratively separate GIS. This work is also aimed at supporting LBS, where content or information is relatively dynamic.

4.1.1 Mobile P2P LBS

Sensing and retrieving network context from scattered domains is redefined as a context-rich location-based service. Figure 4.1 shows, on the left hand side, a naï£¡ve approach which requires a separate query/or update process to each separate domain (or context sensing entity), shown in the figure as separate GIS placed in the same geographic zone. On the right hand side, an overlay constructs a virtual network capable of routing to reach LBS peers (overlay nodes) as well as content. This architecture differs from a classical LBS architecture, in that a P2P relationship exists between back-end GIS. The overlay is used to restructure data, and to route not only to pre-registered nodes, but also to scattered LBS content. The overlay network introduced in Chapter 3 offers the following functionalities:

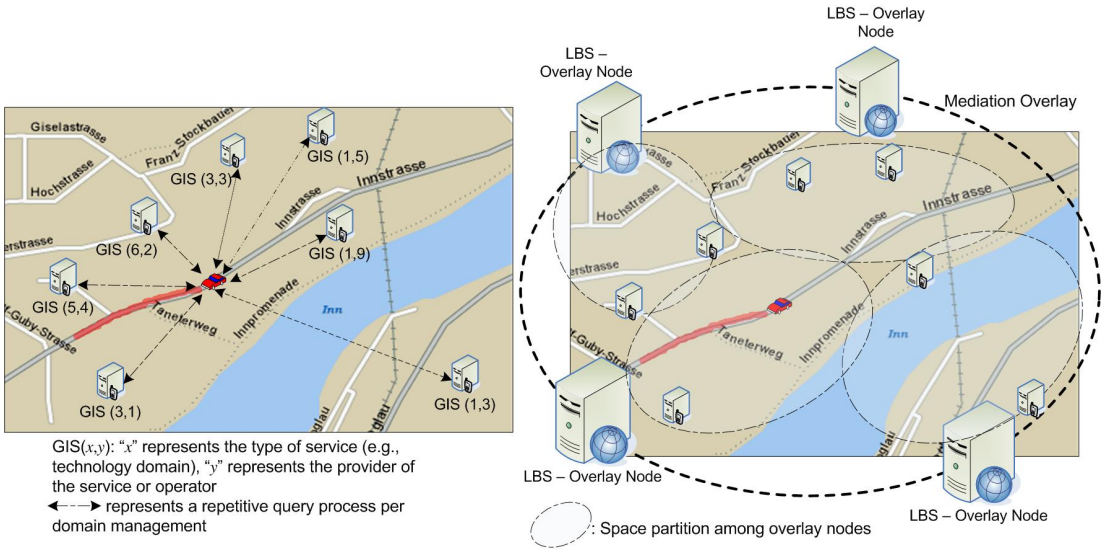

Figure 4.1: Retrieval of network context while on the move - Left: current situation of fragmented LBS among operators and type of services. Right: use of a single management plane to coordinate heterogeneous GIS via an overlay

- **Automatic resource discovery**: While content is pushed by heterogeneous GIS, any other peer can discover this content, which is forwarded back to querying agents in the system (e.g., search peer introduced in Chapter 3).

- **Translation between service providers**: The overlay translates between heterogeneous management domains as it offers an abstraction of objects from their originating GIS and away from the data structure used by the different content providers. The overlay being location-based can compose between different LBS types.

- **Reorganisation of LBS**: A major role of the overlay is to reorganise the scattered information or data domain along the structure of a scalable and well-structured semantic overlay. Independent of GIS domain or operator boundary, the reorganisation of data among overlay nodes requires a unique mapping function. The role of the overlay is to support continuous and spatio-temporal queries for mobile users. This requires an overlay-level addressing of data items, which is independent of their operator or domain addressing or indexing scheme.

Before investigating indexing techniques for a P2P LBS, first, data modelling in centralised systems is explained. Data modelling describes the way data can be indexed and then stored physically.

4.1.2 Data Modelling for Centralised LBS Systems

When talking about a central data base system, the geographically bounded data needs to be structured physically so that a geographic range query takes the minimal search overhead. Since the query is normally specified in a high level search language such as SQL, the search effort then depends on the algorithm which models the data, indexes it, and stores it in the physical database. Historically, B+ Tree index techniques are seen to deliver reasonable response times in large scale relational databases [164, Ch. 3]. In order to preserve the order of multidimensional geographic placement, mapping techniques such as Z-order or Hilbert curves have been used [96; 133]. Despite the fact that there is no ideal mapping technique, in database systems a combination of both some geographic order and a B+ Tree can provide a reasonably good model to order and store geographic data items [164, Ch. 3].

First, an approximation on how to assign a single or multiple keys or indexes to a geographically bounded information is needed. The simplest way to approximate this information is by identifying the smallest bounding square or rectangular box that surrounds the object's geographic position, and then allocate a key to that box. Well-known spatial indexing structures are R-Trees and variants of it. The R-Tree methods are a variation of B+ Trees with the ability to keep proximity between k-dimensional data objects [164, Ch. 3]. Since catering for data proximity is important in both range and nearest neighbour queries, R-Tree methods are, next, looked at into more details.

4.1.2.1 R-Tree Based Indexing

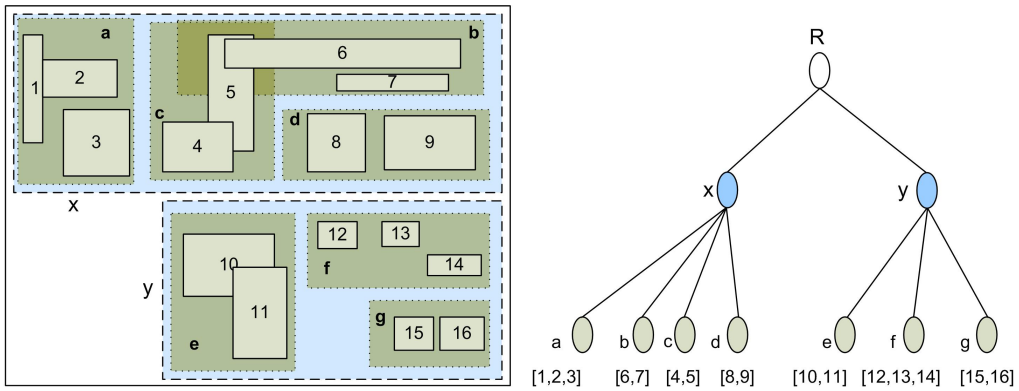

Figure 4.2: A collection of spatial objects and its R-Tree hierarchy - (Source Shekhar et al. [169])

R-Trees apply a hierarchy when ordering data items, which are modelled in Figure 4.2 as approximated boxes which describe some geographic objects such as wireless coverage zones. The indexing scheme is a result of a space filling curve mapping, which gives the order in which these objects should be numbered. The next level of hierarchy follows the same direction (from a to b, to c, to d, etc.), while following the same direction given by some space filling curve. The example shown in Figure 4.2 applies the Z-order Peano curve [139]. Other data models to encode streets and maps exist. But these are out of scope of this work.

Most importantly, when defining a range query, the search starts at the root "R" (in Figure 4.2) where, each branch which further splits at each level is searched and a response is generated at the lowest level of the hierarchy. This type of hierarchical

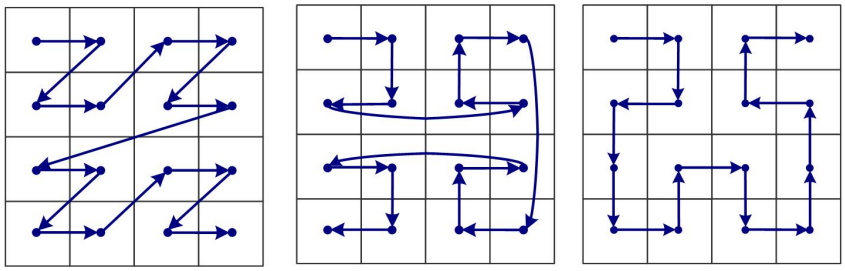

Figure 4.3: Three space filling curve examples - Leaf order in a R-Tree can be determined by the selected space filling curve - from left to right (a) Z-order curve, (b) Grey Code curve, (c) Hilbert curve

structure has been also adopted in hierarchical tree structures used in P2P distributed databases.

Recently, a number of proposals in that direction have suggested to distribute B-Trees on P2P networks [42; 64; 97]. This means that the indexing structure above the leaves, as shown in Figure 4.2 is mapped to separate overlay structures. The mapping between the overlay and the data structure is important in order to construct an ideal semantic overlay, whose ultimate goal is to support range spatial queries.

4.2 Semantic Overlay Structure vs. Data Structure

Dealing with semantic neighbourhood at overlay level has been discussed in Chapter 3. In summary, semantic overlays could be said to be concerned with the placement of data items among overlay nodes so that they are as close as possible to each other in the overlay as they are semantically. This results in minimising the fragmentation of the search requests required to cover a range search or query box.

Originally, DHT protocols have been optimised to solve an exact key match routing problem (called mediation). The DHT maps a key space to a random application level value (or objects), which allows a distribution of data management information in a load-balanced way among the peers. The same applies to the resulting communication or mediation overhead for exact match queries. DHTs, which offer a good way to structure large overlay networks, split a continuous index space rather equally between participating peers, similar to the amount of routing information each peer should store locally. If used unchanged, the mediation effort to launch a range query would first result into a splitting of the query to retrieve each single key contained in that range.

Unless the data is structured in a way that two neighbouring keys on the P2P key space refer to two semantically close data items too, the resulting mediation overhead would quickly explode.

An important aspect of the system requirement is how well can overlays mediate content through location-based range queries. Although range queries in P2P systems have been heavily researched, in the last few years, this research effort mostly focused on partitioning non-uniform object IDs among nodes. In this type of range queries, data ranges are neither continuous nor uniformly partitioned. This is the case for dictionary entries or persons' surnames, where given ranges change abruptly. For this type of queries, several proposals have been made including Kademlia [122] and P-Grid based overlay [57].

The effort to refit the overlay structure to the data ranges results in a clustering of data among neighbouring overlay nodes. The overlay routing effort could be carried out once for each cluster, assuming that each overlay node keeps routing information to its direct neighbours. Reaching each cluster requires the largest lookup cost in terms of overlay hops. Then the remaining neighbouring peers which manage parts of the cluster could be queried recursively. This could be done by mapping data ranges to existing overlay structure without changing the chosen DHT protocol. This approach is illustrated in Figure 4.4(a)

As an example, location-based encoding in the multi-dimensional DHT *content addressable network protocol (CAN)* [155] offers a good alternative to encode 2-dimensional geographic zones (as proposed in [160]). Although offering a good mapping between the multiple dimensions needed to describe geographic zones, CAN based solutions assign the key space in multiple dimensional large zones, leading to flooding at the edge once reaching neighbouring nodes. The dimension conversion is very restrictive, since once set, it is hard to encode anything else apart from zones. Therefore, even for exact match query a range query is actually processed at the edge through flooding [57].

Another approach is done in the reverse manner, which adapts the P2P network structure to the data structure by modifying the DHT to the desired task. The changes are carried out within the DHT protocol to cater for data similarity and file indexing. The CAN approach, where data similarity is loosely mapped to the DHT layer. One way to achieve this is in replacing the hash function that is behind the structure of the overlay and the distribution of objects and data items among the nodes. An example

of this effort is the *locality sensitive hash (LSH)* paradigm, which replaces the DHT's uniform hash [178]. The difference between the two design processes is given in Figure 4.4.

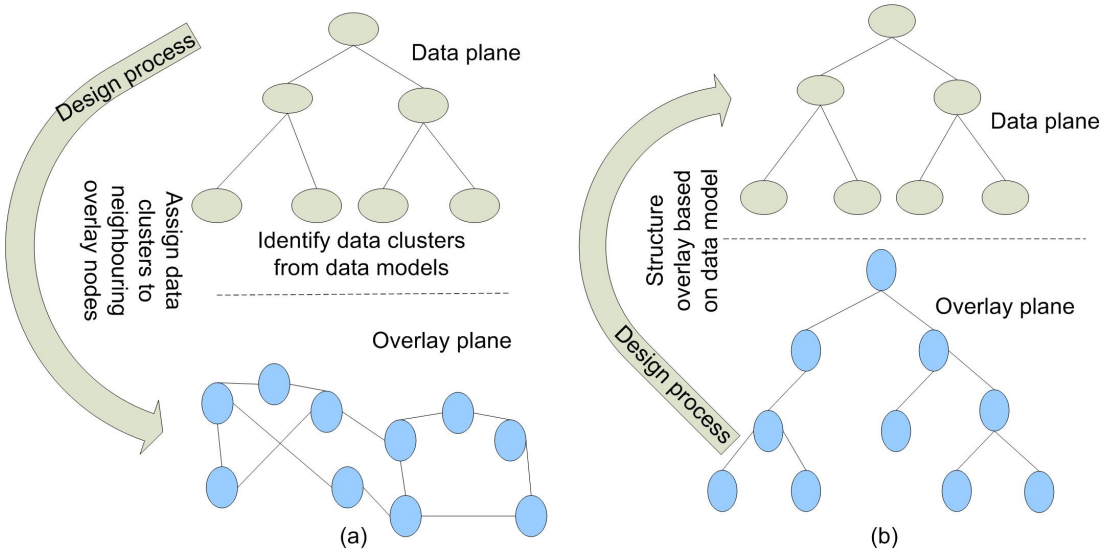

Figure 4.4: Design process of semantic overlays - (a) Data clusters are mapped loosely to overlay neighbours; (b) Overlay structure mimics the data structure for a tight mapping

As an example of the second design process shown in Figure 4.4(b), Datta et al. [56; 57] propose using *prefix hash trees (PHT)* (used in traditional database research) to build a load-balanced binary tree structure, called P-Grid. The prefix relation states that a peer addressed with "001" would be on the route to all nodes or objects, whose addresses share this prefix. Objects are stored only at the leaves of the tree structure, whose depth is determined by the amount of objects at each branch. Once a given load or number of items at a node is reached, the load is split between two new leaves. Combined with search algorithms at the lowest part of the hierarchy, the proposal in [57] is well suited for fragmented keyword based range queries. It is though worth noting that once at the edge the search algorithms are closer to flooding.

Another similar example is given in [50], where a PHT is used to manage geographic entries from the place lab [1] [85]. Objects are assigned to a node, as single keys. When

[1]The place lab [85] records the GPS coordinates of WLAN access points, so that users attached to WLANs can estimate their position.

a certain number of keys is reached, the data space is split further along the chosen leaf requiring additional bits for the new node IDs. The objects have a binary ID of fixed length to describe data's geographic position with high accuracy. However, the geographic ID is obtained from single precise latitude and longitude values, which do not cater for geographic zone indexing.

A range query along a PHT has to involve nodes up the hierarchy, which depends on the size of the range. In the case of frequent queries such as those planned in the mobile case, the nodes higher up the hierarchy would face a relatively high load in broking routes to leaf nodes. In order to optimise the query process near the leaves, Datta et al. [56; 57] suggested to connect leaves with each other in P-Gird, to minimise the need for a query to climb up the hierarchy. The near flooding of leaves also results into a high load, which in the case of mobile users would be required very frequently for each user.

Important to tree-structured systems is their ability in preserving locality, however, they face similar problems to DNS such as high load at the root nodes and vulnerability to node failures, which lead to tree splits.

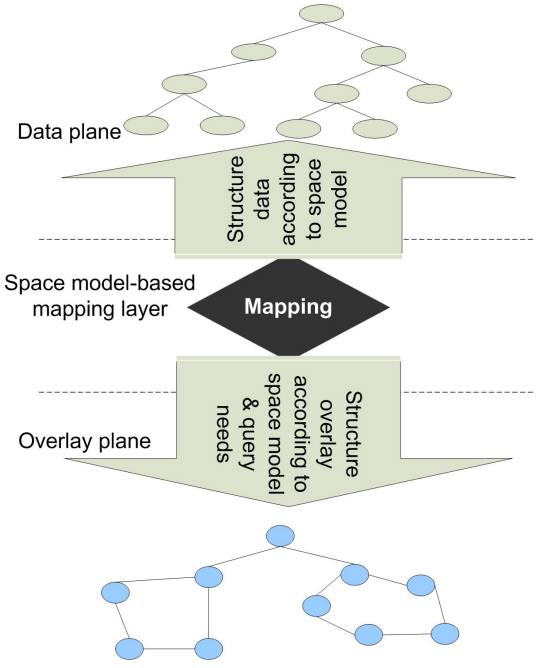

Figure 4.5: Semantic overlay design process based on separating the space model - Both data and overlay nodes are mapped independently to the same space model

In the approach taken in this work, the mapping between data structure and overlay

structure is done in both directions. This is illustrated in Figure 4.5. A space model is proposed, which defines the indexing scheme of a 2-dimensional closed space independent of the objects or nodes mapped into the latter space. This is a layered approach, where data structure could be heterogeneous (not a single indexing scheme). This is already the case when composing between heterogeneous data domains originating from different types of applications and operators. The second mapping is done between the overlay structure and the space model directly. This approach is called a layered approach, as the space model represents a separate layer, which requires one separate and independent mapping process between the data plane and the space model, and another mapping between the overlay structure and the space model.

The next part investigates the requirements on the space model, once a DHT is selected. Since layering also means separation of concern, any other DHT could be used at the overlay layer, with some structuring changes resulting from the space model.

4.2.1 Overlay Requirements for Range Queries

DHT-based overlays are network graphs, which are concerned with topological scalability. The various DHT protocols differ in their graph structure in terms of *(i)* their *diameter*, which gives the upper bound of hop-count needed for a single lookup operation, and *(ii)* their *degree*, which determines the size of the routing table stored at each node [178]. Several protocols such as Chord, Pastry, and Tapestry are based on the Plaxton Mesh [142], which achieves $(\beta - 1)log_{\beta}(N_p)$ diameter and $log_{\beta}(N_p)$ degree (for N_p number of nodes forming the network). Here β indicates the base of the DHT identifier space, for example $\beta = 2$ in Chord. Another family of DHTs is for instance the CAN protocol, which manages a d-dimensional space with a degree $2d$ and a diameter $\frac{1}{2}d(N_p)^{\frac{1}{d}}$. Chord is a DHT protocol, whose diameter leads to one of the most scalable search efforts. The degree of each node in Chord results in a logarithmic cost of maintenance. The latter refers to the amount of state managed at each overlay node in order to maintain the overlay structure. A focus is made in this work on a Chord-based DHT layer. CAN or other DHTs can be used as well, but the mapping function has to be adapted for each protocol.

Assuming the existence of a perfect data and node structure resembling a data R-Tee, the approach advocated in the remaining of this chapter is to allow nodes as close as to the leaves to connect to each other, based on a ring structure using the Chord

protocol. The range queries needed in a mobile LBS involve small geographic ranges, and therefore, should only involve those nodes close to the subspace in which the range query is defined and started from. Nodes higher up in the hierarchy, are only needed for larger range queries or remote queries. Further need for structure or hierarchy in order to reduce maintenance cost are also investigated.

For the purpose of analysing geographic range querying among overlay based solutions, the following methodology is needed:

1. The diameter of the spatio-temporal queries is studied analytically. The analytic analysis results into design principles which can be applied at the design phase of structuring semantic overlays based on a space model, on the one hand, and to the data plane, on the other. The result should be a self-organising maintenance protocol to fit the structure of the overlay to offer efficient routing and maintenance.

2. In order to verify the results of the numeric results, a simulation is developed. The simulation study should also verify the design principles and the design process used to construct an efficient overlay semantic network.

3. The user behaviour can be best modelled through simulation. The simulation model recreates a search scenario based on some movement pattern. Adapting the query size to movement is needed to investigate the scope of the query and the overhead linked to it. After, mapping the numeric model's parameters to those of the simulated scenario the simulation results are used to also validate the analytic model.

The use of a modified Chord DHT layer is shown to deliver a satisfying and self-organising solution for mobile spatio-temporal queries.

4.2.2 Semantic Querying Using Chord

Chord uses a uniform key space represented by an integer set $S_{addr} \subset \mathbb{N}$ comprising number ranging between 0 to $2^m - 1$, where m is an integer representing the number of bits used to describe the size of the Chord ID space. Both objects and nodes are addressed by an associated key or a subset of keys. For a well balanced Chord ring

consisting of N_p nodes, each node j should manage a key subset $S^j_{addr} \subset S_{addr} | j \in \{1, 2, ..., N_p\}$, which distributes the 2^m keys in a load-balanced manner among the peers.

Using Chord without change means that each object is assigned a randomly distributed key to address it. The query type targeted in this work would search for objects that are semantically close to each other. If used unchanged, the search is said to be naïve.

Search complexity following a naïve approach Assuming N_p nodes in a stable Chord ring, a search for a single key requires $O(log(N_p))$ messages. A search query involving W semantically neighbouring objects in a uniformly structured Chord ring require one search for each of the W keys in the query box resulting in

$$O(W * log(N_p))$$

messages.

Instead of looking up each key separately, clustering items along the ring, so that their semantic neighbourhood is preserved by the addressing scheme, reduces the splitting of the mediation effort. The $log(N_p)$ search effort should be repeated for each cluster instead of each object. Given that a cluster within a range of size W gathers w objects, the search effort is reduced to $O((W/w) * log(N_p))$ messages. This assumes that W-large range is split among uniform and continuous clusters (along the ring). If a cluster (mapped to uniformly increasing neighbouring keys) is split among M direct neighbouring peers, then the overhead becomes:

$$O((W/w) * [log(N_p) + M]) \tag{4.1}$$

In order to achieve the sought distribution of data items among the peers, the following functions in a Chord like DHT need to be modified:

- Inserting an object in the overlay requires a coding function from the two-dimensional geographic coordinates to a single coordinate, which preserves geographic neighbourhood at the single-dimensional ID space.
- Query definition depends on a similar transformation function transforming a geographic zone (e.g., square, circular, or rectangular surface) into the set of Chord keys (or preferably Chord key ranges).

- A set of neighbouring keys in the Chord space are easily converted back into a geographic range.

This transformation of a geographic information into a single-dimensional key space is called *"Geocoding"* [164].

The problem with Chord is that its query language requires a precise known object ID that can generate a precise $(key, value)$ pair, making range queries difficult. The space modelling is needed to achieve an efficient transformation of geographic information into clustered Chord keys.

4.3 Space Filling Curve Based Geocoding

Geocoding is the process of assigning geographic identifiers (e.g., ZIP code or street name to geographic coordinates (such as latitude and longitude) in order to map a given feature or a given data to its geographic position. The identifier is used to then associate the coded element with a geographic information server (GIS) [164].

4.3.1 Layered DHT for Geocoding

The approach selected in this work is to have a layered approach of coupling between decentralised GIS systems and their underlaying overlay structure used for mediation and content discovery. Assuming that the selected DHT is Chord, the overlay protocol layer offers a basic networking service through an overlay structure, which, in a first instance, is left unchanged. The overlay layer also manages an address space of 2^m 1-dimensional keys. Geocoding takes place between the data structure layer and the overlay layer, and maps geographic information (such as latitude, longitude) to the Chord binary key space. This has to take place in both directions.

In operating the layered approach, a node can identify an object first through its geographic attributes. The geocoding function then maps geographic attributes of an object o_x into the 2^m key space. The object o_x is then stored at the node or nodes which are responsible of the mapped keys.

In the layered approach, the original Chord is considered to run as protocol layer below the hashing layer. When compared with the classical Chord protocol the geocoding layers takes in charge the following functionalities:

- Skip transformation used by the hash function in Chord so that the 2^m geographic keys are not randomly distributed among Chord nodes.
- Apply a similarity rule to address peers or nodes which preserves geographic neighbourhood.
- Query definition at the geocoding layer transforms a geographic range into the set of keys which need to be searched for associated data.
- A set of neighbouring keys in the Chord space are easily converted back into a geographic range

Next, the geocoding and reverse geocoding based on the Hilbert space filling curve are addressed.

4.3.2 Peano Space Filling Curves

Space is approached as a separate entity, which is filled by both objects and nodes following some indexing scheme. A closed space can be split into zones or subspaces. The ideal indexing scheme should reflect the density of a given zone or subspace. Objects and nodes are then linked to an index that represents its position in the space. An indexing scheme performs well, if the difference between two index values reflect the Euclidean distance between the objects or nodes mapped to that space. Such functionality is offered by space filling curves, also known as Peano curves.

Peano [139] discovered the existence of a continuous curve which passes by every point of a closed square. Jordan's precise notion of continuous curves, redefined a Peano curve as a continuous mapping of the closed unit interval $I = [0, 1]$ into the closed unit square $S = [0, 1]^2$ [130].

Definition 4.3.1 *If a mapping $f : I \rightarrow E^n (n \geq 2)$ is continuous and $f(I)$, the image of I under f, has positive Jordan content (area for $n = 2$ and volume for $n = 3$), then $f(I)$ is called a space filling curve. E^n denotes an $n-$dimensional Euclidean space [130].*

Hilbert analysed the geometric construction of the space filling curves. If the interval I can be mapped continuously onto the square S, then after partitioning I into four congruent subintervals and S into four congruent subsquares (S_0, S_1, S_2, S_3), each

subinterval can be mapped continuously onto one of the subsquares. If this is carried on infinitely, I and S are partitioned into $2^{2\gamma}$ congruent replicas for $\gamma = 1, 2, 3, 4, ..., \infty$.

The Hilbert curve is a special space filling curve with the geometric form illustrated in Figure 4.6. In general, a $d-dimensional$ Euclidean space with finite granularity can be filled with the $\gamma - approximation$ of a $d - dimensional$ Hilbert space-filling curve, which maps an integer set $[0, 2^{\gamma * d} - 1]$ into a $d-dimensional$ integer space $[0, 2^{\gamma} - 1]^d$.

Definition For γ, let H_{γ}^{d} denote the γ order approximation of a $d - dimensional$ Hilbert space-filling curve, which maps $[0, 2^{\gamma * d} - 1]$ into $[0, 2^{\gamma} - 1]^d$.

4.3.3 Modelling Closed Geographic Spaces

When defining a geographic space, this is seen as Euclidean 2-dimensional zone mapped to a square mapping whose sides are given in some geographic unit (metres or kilometres). The largest square mapping has to cover a surface equal to that of the earth surface, which is given as 510,072,000 km^2 [14]. However, since the earth is a sphere, it is hard to map the earth surface to a bounded flat square space. This refers to the old cartographer's dilemma of mapping a sphere surface to a flatted map. Here, several alternatives can be found. The worst flattening technique would be map the sphere to a bounded rectangle whose length equates the diameter of the sphere at the equator and its width is the distance between the two poles. The resulting map of the world keeps proportionate scaling at the equator while it distorts the horizontal distances near the poles. The distorted earth surface is then 800,000,000 km^2, which is about 289,928,000 km^2 larger than the spherical surface.

Other possibilities include mapping the earth to a cube, which is easily mapped to a continuous Hilbert curve (as given by Dennis in [61]).

For an analytic model, it is sufficient to look at the surface of a closed geographic space modelling the surface of a large and heavily populated country. As an example Germany could be modelled through a $2^{2\gamma}m^2$ grid surface. Mathematically that is choosing the right grid mapping, which represents sufficient granularity to describe $1m^2$ surface or $1m$ distance. The surface of country like Germany is about 300,000 km^2. The maximum number of grid cells is given as $2^{\lfloor log_2(Surface\ in\ m^2) \rfloor} = 2^{38}$. In other words, it is sufficient to map the surface of a country like Germany by a 2^{38} cells of $1m^2$ each. The required Hilbert curve can be given as H_{19}^{2}.

4.4 Hilbert-Based Indexing Scheme

In this part, the size of the selected identifier and its dimension are investigated.

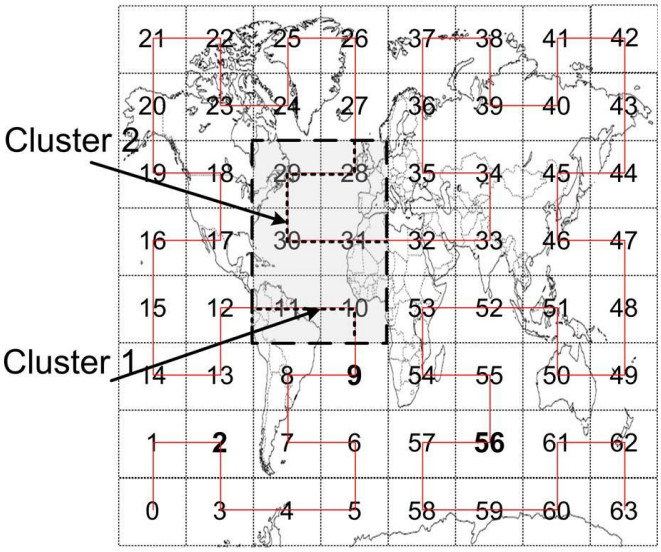

Figure 4.6: Geocoding using the Hilbert space filling curve H_3^2 ($\gamma = 3$, and $d = 2$) - $[0, 2^6 - 1]$ integer coordinates into the $2-$*dimensional* grid mapping covering the earth surface $(x, y) \in [0, 2^3 - 1]^2$

In Figure 4.6 the 3^{rd} approximation of a $2-$*dimensional* Hilbert curve is taken to model the earth surface (i.e. $\gamma = 3$). This means that given that the earth surface is covered by a $2^3 \times 2^3$ mapping (shown on the left), any point of the earth could be addressed with the coordinates of the bounding box in which this point exists. The Hilbert curve mapping converts a 2-dimensional (x_n, y_n) coordinates to a 1-dimensional key (i.e. a single integer). The example shows in Figure 4.6 how the coordinates of the zones $(1, 1)$, $(3, 2)$, and $(5, 1)$ are converted to integer IDs 2, 9, and 56 respectively. The mapping used to divide a space in a $2^\gamma \times 2^\gamma$ grid requires a H_γ^2 Hilbert curve, which converts the $2^{2\gamma}$ possible (x_n, y_n) coordinates into one of the $2^{2\gamma}$ integer IDs. Such an integer space requires 2γ-bit long ID. The addressing of each geographic zone inside a single cell results in a binary address which depends on (i.e. the approximation) and the (x_n, y_n) coordinates. The same transformation could be done in reverse.

4.4.1 Modelling Geographic Information Using the Hilbert Curve

It can be concluded from the above sections that it is possible to encode geographic information in general on a Chord ring using a Hilbert transformation of 2-dimensional ID space into a single ID space. The granularity of the modelled space can be analysed theoretically to identify conversion parameters (e.g., the degree of the Hilbert space, which designs the amount of geographic information encoded in the binary Chord key space). Similar to the measurement of the coast of England, the modelling of the earth surface into a grid of neighbouring square cells is a fractal problem. The fractal nature of Peano curves is reflected by the binary construction of keys.

Fractal Nature of the Hilbert Addressing Assuming a $2^\gamma \times 2^\gamma$ mapping of a 2-dimensional space is filled by the γ-approximation of a Hilbert curve, if each cell in that grid is divided into 4 subsquares, this is the $\gamma+1$ approximation of the same space which addresses the new cells with 2-bit longer keys. This process can be repeated indefinitely resulting into a fractal.

4.4.2 Clustering in the Hilbert Curve

When looking at query definition at the data plane, a search range is first defined as geographic zone which can be bounded by several concatenated smaller rectangular boxes. The latter boxes are drawn geometrically, so that they cover the geographic zone in which the query is defined. They are also defined through the mapping I which represents the grid system in which the earth surface has been mapped.

Taking any square box inside that range, the query can be split into several clusters which could be compared with the number of leaves at the lowest level of an R-tree hierarchy, when looking at the data structure. The difference here, a query is defined over all possible leaves which exist in the data plane instead of only those leaves that include objects. A square box covered by the range placed on the grid mapping can be said to contain $2^k \times 2^k$ possible keys, while $k < \gamma$. This translates into a search box which needs to check every cell covered by the mapping I modelled to cover the earth surface or a portion of it.

Clusters are groups of grid points inside a query box that are consecutively connected by a space filling curve which maps the modelled surface in the grid I. An example is

given in Figure 4.6, where a random query box is selected. Two clusters exist in that query box $[10, 11]$ and $[28, 31]$.

The authors in [130] have produced a closed formula for the average number of clusters in a $2^k \times 2^k$ square query box positioned randomly within a $2^{k+n} \times 2^{k+n}$ grid mapping of a 2-dimensional space:

Definition 4.4.1 (Average number of clusters) *Given a two-dimensional grid partitioned space of the size $2^{k+n} \times 2^{k+n}$, and a search box of the size $2^k \times 2^k$. Then the average number of clusters resulting from a Hilbert filling curve is*

$$N_2(k, k+n) = \frac{(2^n - 1)^2 2^{3k} + (2^n - 1)2^{2k} + 2^n}{(2^{k+n} - 2^k + 1)^2}$$

which is asymptotically bounded by the value 2^k (i.e. $\forall (n, k) \in \mathbb{N}^2$, where $n+k = \gamma \wedge k < \gamma$, $N_2(k, k+n)$ approaches 2^k) [130, Theorem 2]. The list of parameters are listed in Table 4.1.

Parameter	Definition
$\gamma = n + k$	The granularity of the modelled space given as the maximum index of both x and y coordinates where, $(x, y) \in [0, 2^\gamma - 1]^2$ and $(x, y) \in \mathbb{N}^2$
k	The binary index of a square search box inside covering the area of $2^k \times 2^k$, where $k < \gamma$
H^d_{n+k}	A 2-dimensional Hilbert curve filling a $2^{k+n} \times 2^{k+n}$ grid covered 2-dimensional space
$N_2(k, k+n)$	The average number of clusters inside a $2^k \times 2^k$ search box placed somewhere inside the closed space of size $2^{k+n} \times 2^{k+n}$ and filled by a H^d_{n+k} Hilbert curve
N_p	Total number of peers taking part in the Chord network
M	The number of peers covering a single cluster (worse case)

Table 4.1: List of the main parameters used in the asymptotic study

When compared with other space filling curves in terms of their clustering property, the Hilbert Curve performs best [130]. Knoll et. al. [106] have shown that out of the Peano space filling curves, the Hilbert curve maps geographic neighbourhood best. In other words, given two neighbouring binary addresses obtained following a Hilbert

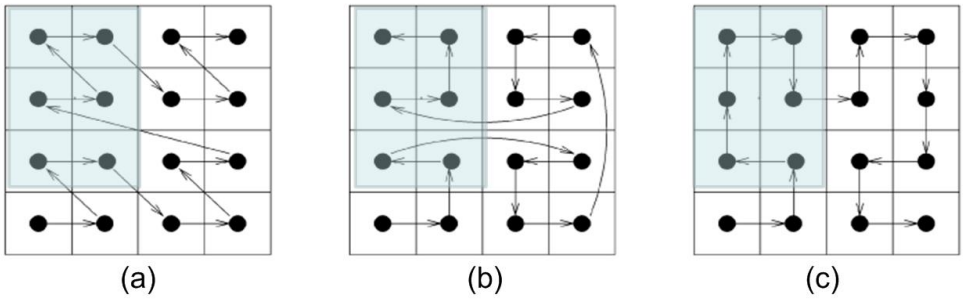

Figure 4.7: 2-dimensional second degree (i.e. $\gamma = 2$) Peano space-filling curves - (a) reverse Z-curve, (b) Gray-coded curve, (c) Hilbert curve

curve, these two points are geographically closest compared to other space filling curves. The example shown in Figure 4.7, shows three alternative space filling curves covering a $2^2 \times 2^2$ mapping. For the query box placed at the same position for all three examples, It can be seen that whereas in the Z-order curve and the Gray-coded curve there are two clusters inside the same query box, in the Hilbert curve, only one cluster is identified.

4.4.3 Addressing Objects in a Chord Ring

Mapping Hilbert generated keys to a Chord ring is about mapping a set of the Hilbert generated ID space, which describes 2-dimensional geographic data with the help of $2^{2\gamma}$ binary keys (depending on the granularity of the data space), to the 2^m Chord key space. It is first assumed that the data space keys are mapped one-to-one to the Chord keys, which is the worst possible mapping scenario, since the granularity of the data space is then kept one-to-one in the Chord key space. Later in this chapter it can be shown how adapting the number of data keys mapped to a single chord key could be done. Further analysis of the mapping between the key space and actual Chord nodes is also analysed.

In the next sections the complexity of a geographic query is estimated analytically. The result should indicate the implementation parameters that need to be optimised in order to model the earth surface as a whole then, more specifically, an urban area, and finally some heterogeneous wireless cells. The range query is carried out according to the following steps:

1. First the clusters within the given query box need to be identified.

2. A $get(key)$ message is sent to reach the peer responsible of the first key within that cluster iteratively.

3. Hopping between peers covering the cluster follows recursively.

This translates into a $get(range)$ call for the modified clustered Chord addressing scheme.

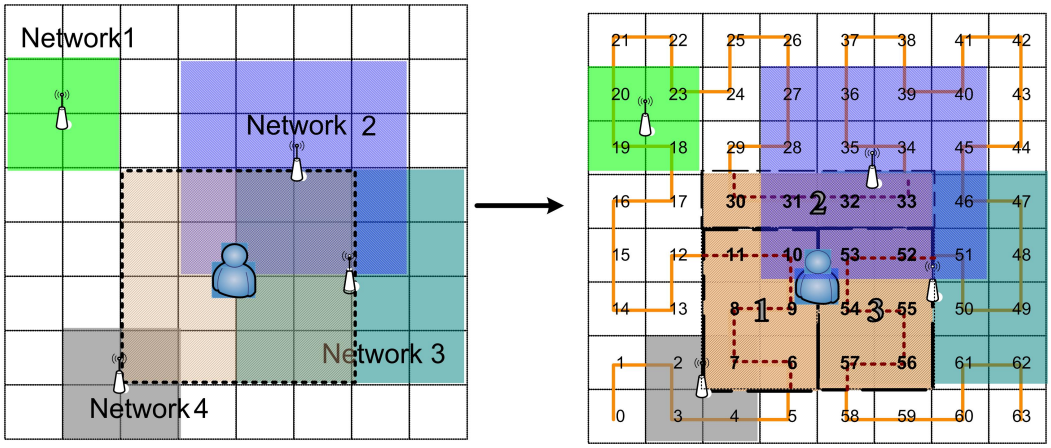

Figure 4.8: Hilbert curve-based transformation from 64 (x, y) coordinates to 64 one-dimensional 6-Bit IDs - $d = 2$ and γ=3

The example shown in Figure 4.8 illustrates the transformation of a 64-cell grid space addressed with (x_n, y_n) coordinates, where $(x_n, y_n) \in \mathbb{N}^2 \mid 0 \leq x_n \leq 7$ and $0 \leq y_n \leq 7$. The resulting $2^3 \times 2^3$ mapping converts 2-dimensional ID into 1-dimensional $3 + 3 = 6$ bit long keys. In the shown example (Figure 4.8), there are 4 access cells requiring each a different number of keys to describe their geographic coverage. As an example "Network 1" (in Figure 4.8) requires four keys 18, 19, 20, and 23. The clustering effect is demonstrated by the given query box in the middle. To start the query, the start of the cluster and its length need to be identified. Here, three clusters exist ($[6, 11]$, $[30, 33]$, and $[52, 57]$), requiring each a separate $get(range)$ call. Identifying the peers responsible of each cluster is bounded by the overlay routing cost $log(N_p)$ (where N_p is the number of peers part of the Chord ring). Now instead of using the Grid space, the 64 IDs are mapped to a ring, starting with key 0 and finishing with key 63. This ring is the logical illustration of a simple Chord ring (see Figure 4.9):

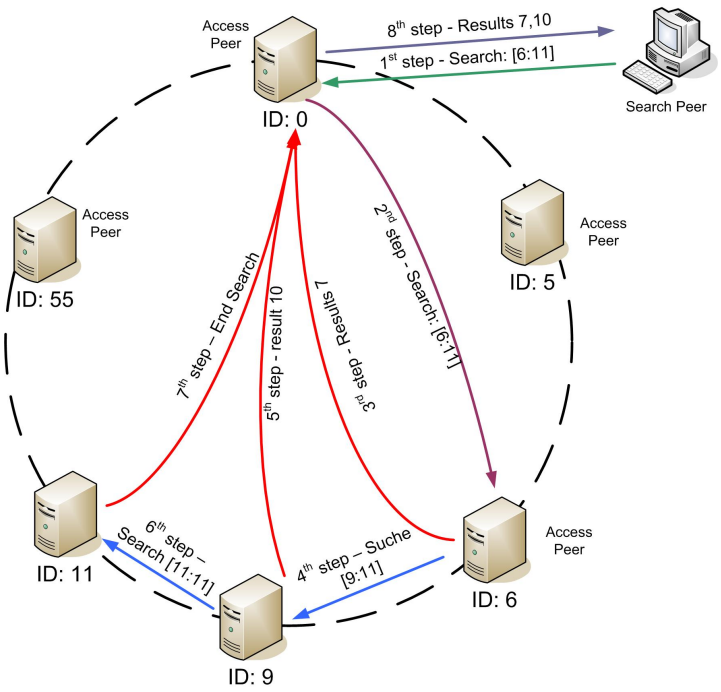

Figure 4.9: Search for cluster 1 (from ID 6 to ID 11) - shown in Figure 4.8

- Step 1: the search initiator defines the search criteria and sends a request to an access peer to start a query on the Chord ring. In Figure 4.9 a single range is queried between key ID 6 and 11 (i.e. one cluster in the Hilbert transformation of the query box in Figure 4.8).

- Step 2: the peer which receives the request from a search peer, starts a range query session, which calls a $get(range)$ method to reach the node responsible of the first key in the range. In Figure 4.9 the peer with ID 6 is reached after a logarithmic routing effort $O(log(N_p))$, where N_p the number of participating peers). In the figure, a direct finger existed in node 0's finger table.

- Step 3: the peer with ID 6 responds with objects found. In Figure 4.9, the latter peer is only responsible for objects addressed with keys 6, 7, and 8. These objects are sent back to the node with ID 0. If no objects are found the peer with ID 6 doesn't send any messages back since the cluster or range is not yet covered.

- Step 4: if the range is not covered, the query is further forwarded to the neighbouring nodes in a recursive manner until the end of the range is reached. Step 3

is repeated by each node, where objects are found. This recursive routing occurs $O(M)$ times, since it could be said that each Hilbert cluster is always managed by peers whose IDs consists an array of successor nodes. For worst case estimates, each Chord ID is managed by a separate node, then M is said to be equal to the length of the cluster (subtracting the node heading the cluster) (e.g., the range $[6, 11]$ has $M = 5$, if each single key was managed by a separate node).

- Step 5: if the *get*(*range*) reaches the last ID in the range, then the query is finished, indicated with an explicit response by the node managing the last key. The query session can be assumed to be finished.

- Step 6: the results of the query are sent progressively back to the search peer (as soon as any peer responds) rather than once after the results are gathered by the node with ID 0.

Search complexity for each single cluster Assuming N_p nodes in a stable Chord ring. In order to query a single cluster

$$O(log(N_p) + M)$$

messages are needed.

4.4.4 Query Complexity for a Whole Query Box

In addition to the logarithmic cost to reach the head of a given cluster, the partition of a cluster among peers is a major contributor to the query cost. The value of M indicates the fragmentation of a cluster among neighbouring peers. This depends on the granularity of the space model taken and therefore the required degree γ of the Hilbert curve, and then on the partition of the maximum possible keys among peers. Assuming that in the worst case each key is associated with a different peer, the value of M_{max} can then be estimated as the length of each cluster inside a square query box. Formally this becomes:

$$M_{max} = \frac{total\ number\ of\ keys\ inside\ a\ query\ box}{average\ number\ of\ clusters\ inside\ a\ query\ box} - 1 = \frac{2^k \times 2^k}{N_2(k, k+n)} = O(2^k)$$

Definition 4.4.2 (Message complexity of the whole search box) *Assuming N_p nodes in a stable Chord ring and each node only manages one key (which is its own ID). A search query of the size $2^k \times 2^k$ in a uniformly structured Chord ring will require for all 2^{2k} keys in the query box*

$$O(N_2(k, k+n) * (log(N_p) + M_{max})$$

messages. The maximum length of the cluster after reaching the peer heading the cluster is M_{max}.

$$M_{max} = \frac{2^k * 2^k}{N_2(k, k+n)} - 1$$

Combining the two results, the complexity of a $2^k \times 2^k$ range query in a Hilbert curve partitioned $2^{k+n} \times 2^{k+n}$ space, and managed through a one-dimensional Chord ring results in the following complexity.

$$O(2^k * [log(N_p) + 2^k - 1)]$$

4.5 Analytic Study of Query Behaviour in Chord

Numerical estimation of query behaviour is made possible due to the deterministic way the Hilbert curve can cluster and partition space among Chord nodes. The numeric communication bound could be used to further analyse the design parameters needed for efficiently indexing geographically linked wireless networks.

4.5.1 Data Granularity Effect on Query Overhead

The query behaviour as given by the result in Definition 4.4.2 is compared for different granularities (γ) of the fractal representation of geographic areas. Figure 4.10 shows the effect of choosing the right Hilbert curve degree on the overhead of a range query. The flattened square surface of $300,000km^2$ could be modelled with different cell-sizes translating in different Hilbert curve degrees ($\gamma = k + n$, where the size of each unit cell is $\frac{300,000km^2}{2^{2 \cdot \gamma}}$). The cases shown in Figure 4.10 are those of $(k+n) = 19$, 18, 17, 16, 15, and 14 resulting in a Chord key space of 38, 36, 34, 32, 30, and 28 bits respectively. The most thorough mapping is assumed, where each key in the data model is mapped to each Chord key, which means the Chord key space consists of $2^m = 2^{2\gamma}$ keys. It is

also assumed that each key is assigned to a peer to allow worst case estimate as given in the formula in Definition 4.4.2

$$O(N_2(k, k+n) * log(N_p) + M_{max})$$

.

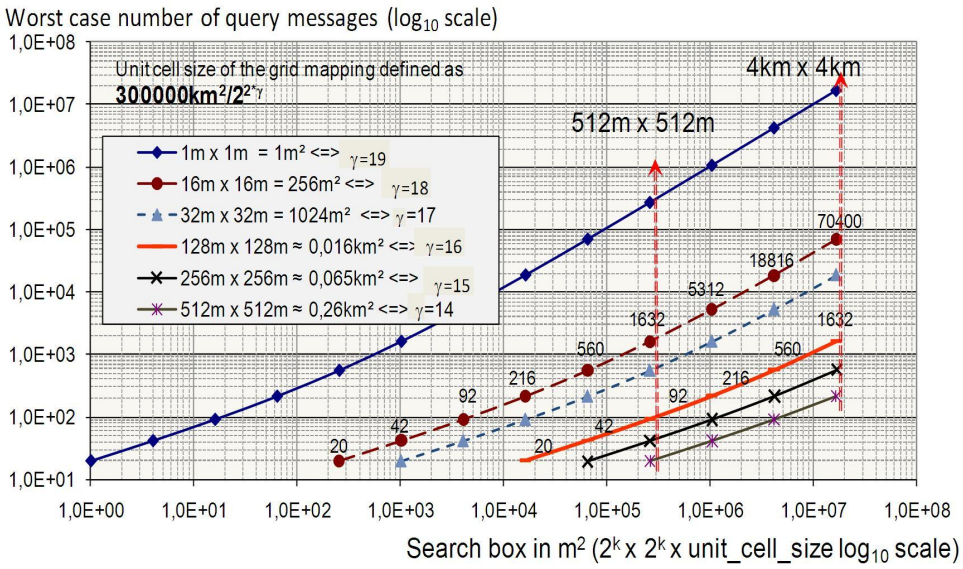

Figure 4.10: Estimation: varying query box for different Hilbert granularities - (log scale)

The geographic size of each grid cell is given in the legend of Figure 4.10. The query box (see Table 4.2) is varied from $k = 0$ to $k = 12$, resulting in a different geographic range depending on the granularity of the data model. The more granular the Hilbert curve is, the larger the overhead becomes. This is partly due to the fact that each object (in this case a wireless cell) of let say $500m \times 500m$ size requires 2500 Chord keys when using $\gamma = k+n = 19$, and only one Chord key when using $\gamma = k+n = 15$. The problem with $\gamma = 15$, though, is that a smaller cell of let say $50m \times 50m$ will be encoded as well in a $500m \times 500m$ cell adding a significant loss of information about the stored data in the Chord ring. During the encoding of data items, it is most efficient to limit the number of keys per data item, but not to loose the granularity of the heterogeneous coverage (i.e., the ability to differentiate between micro, mini, and macro wireless cells). In fact in the simulation scenarios, $\gamma = k + n = 15$ is chosen leading to a granularity of

16m side distance, which means a 50m radius WLAN cell would surround a square cell of surface $2 \times 50^2 m^2$ and requires about 16 bounding squares of area $19^2 m^2$ each. In other words, 16 Chord keys are required to represent the granularity of the above data item.

4.5.2 Prefix-Based Aggregation of Data Addressing Among Peers

Another optimisation aspect is in the distribution of subsequent data IDs among Chord nodes. In this case the unrealistic worst case assumption of assigning each key to a separate node is eased. A peer is assumed to manage a considerable number of keys, and the objects corresponding to those keys. This translates into assigning several data keys to a given peer. The Hilbert transformation offers a prefix relationship which could be utilised in this assignment.

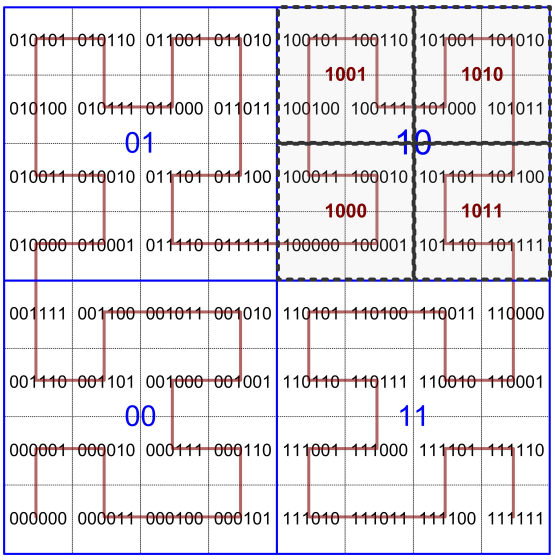

Figure 4.11: Hilbert prefix relationship - $\gamma = 3$ results in data keys 2γ-bit long, first prefix: top left $d = 4$ aggregates $2^{2\gamma-d} = 2^2$ keys together, second prefix: $d = 2$ aggregates $2^{2\gamma-d} = 2^4$ keys together

If the data space is described with IDs of length 2γ-bits, and the Chord ring manages a key space of m bits long IDs, where $m \leq 2\gamma$. The number of peers populating the Chord ring is N_p. In order to preserve the load balancing rule of Chord, we assume an equal distribution of the 2^m keys among the N_p peers, where $N_p \ll 2^m$.

Similarly, the data ID space is assigned in a load balanced manner to the N_p nodes. If we assume that the number of peers is selected so that $N_p = 2^d$, where $d < m \ll 2\gamma$. It could be also shown that there is a prefix relationship between keys assigned to each node and the Chord node ID as well. The prefix is d long and covers the most significant bits of any Chord node and its assigned range.

Therefore, each peer is responsible for $2^{2\gamma-d}$ keys at the data space (or 2^{2m-d} if several data plane IDs are assigned to a single chord key), where all objects managed by the same peer share the prefix which is d long, and $d < m \leq 2\gamma$.

An analytic example is given in Table 4.2. The data ID space describing the 300,000 km^2 flat square surface, can be modelled through a $\gamma = 19$ in 2^{38} grid cells of about $1m^2$ surface each. The unique 1-dimensional key obtained through the Hilbert mapping is further mapped to the same key in Chord (i.e. $m = 2\gamma$). The Chord space manages $2^m = 2^{2\gamma}$ possible keys. A masking of the Chord keys allows to remove some of the granularity of the data space from the Chord key space. Assuming that now the Chord ID space is distributed among $N_p = 2^d$ peers, where $d = 20$ (in Table 4.2), the number of data keys per Chord peer becomes $2^{2\gamma-d}$. There is a natural partition of the data space keys among the peers along a prefix of length $d = 20$. Each node covers a geographic range equivalent to $\frac{300{,}000km^2}{2^{20}} = 0.534 \times 0.534km^2$.

Parameter Setting	**Geographic Interpretation**	**Chord Execution**
Modelled Surface	$300000km^2$	Total modelled space
$2^\gamma \times 2^\gamma$	Number of cells in a grid covering the earth surface the size of each cell is $S_\gamma = \frac{3\times10^{11}}{2^{2\gamma}}m^2$	For a H^2_γ Hilbert Curve 2γ-bit long Chord key length is required
$N_p = 2^m$	Each peer is responsible $2^{\gamma-m}$ unit cells	Each peer manages $2^{\gamma-m}$ keys
$2^k \times 2^k$ query box	Each query box covers an area of $2^{2k} \times S_\gamma = \frac{3\times10^{11}}{2^{2(\gamma-k)}}m^2$	Each query box requires $N_2(k, k+n)$ $get(range)$ calls
M	The number of additional peers involved in retrieving each cluster	Thanks to prefix masking $M \ll M_{max}$

Table 4.2: Complexity reduction through possible optimisation

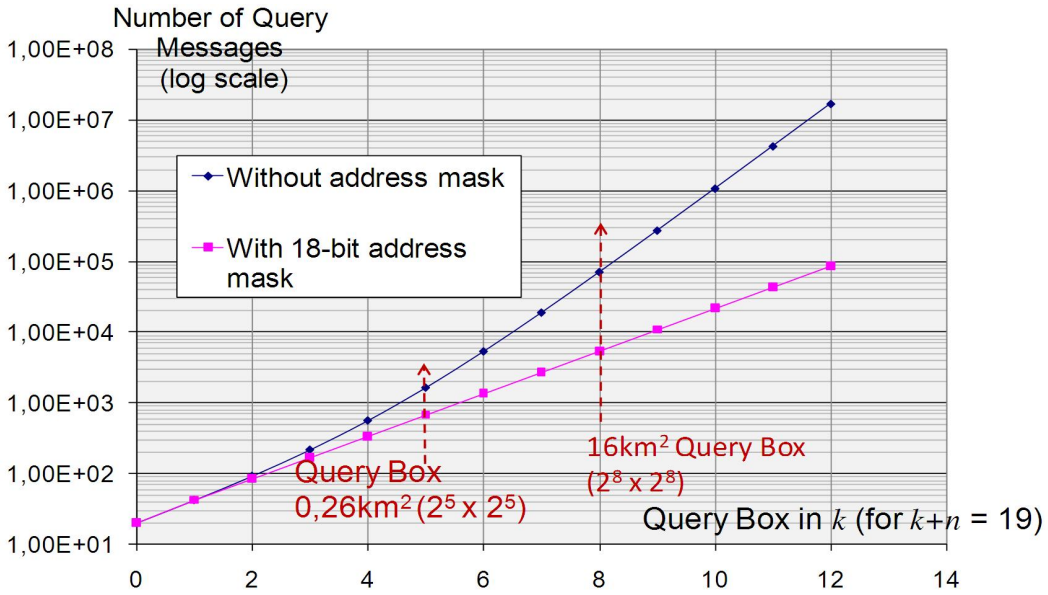

Figure 4.12: Search complexity ($\gamma = k + n = 19$ and comparing worst case with prefix masking for $m = 20$ - With $m = 20$ all keys within the scope of a given peer share a $38 - 20 = 18 - bit$ long node IDs

Figure 4.12 demonstrates the optimisation effect of distributing the 2^{38} key space among 2^{20} peers. The query is carried out for the search box $2^k \times 2^k = 2^{20}$ representing a surface of about $1km^2$.

The partition of the modelled clusters could be optimised. The number of additional peers besides the first node heading the cluster is given as M.

Search complexity after partitioning keys through common prefix Given that a $2^{k+n} \times 2^{k+n}$ grid covered space, a search box of the size $2^k \times 2^k$, and $N_p = 2^d$ nodes forming a stable Chord ring, where the number of Chord keys separating each two peer IDs is $\frac{2^{k+n}}{N_p}$. The estimated Chord complexity is given as mentioned in Definition 4.4.2 as

$$O(N_2(k, k+n) * (log(N_p) + M))$$

Where M is now estimated as the number of additional peers that are needed to cover a given cluster. Depending on the number of keys assigned per Chord node and assuming a prefix relationship common between every node and its assigned binary ID space of

length d most significant bits of the ID space.

$$M = \begin{cases} 0 & , \text{ if} \frac{2^{2k}}{N_2(k,k+n)} - 1 < 2^d \\ \left\lfloor \frac{2^{2k} - N_2(k,k+n)}{N_2(k,k+n) \times 2^d} \right\rfloor & , \text{ otherwise} \end{cases}$$

The above strategy in partitioning the object key space among the peers is easier to analyse. It associates a key space on a higher granularity to a less granular peer ID space, while preserving geographic aggregation and clustering between the two. The problem with that partition is that it is unaware of the actual number of objects associated with each key. For instance in urban centres, the number of wireless access networks and their overlapping coverage results in a higher density of the number of data objects associated with each binary key.

Another criteria for assigning keys among nodes is through progressive discovery of continuous ranges and their density of keys. It is also important to partition the number of objects stored by neighbouring peers so to achieve load balancing.

4.5.3 Modifications of the DHT Ring Structure

Another important aspect is the need for hierarchy, which has been demonstrated by the use of prefix-based aggregation. Here it is questionable that a ring structure covering the whole earth surface is really necessary.

The need for hierarchy can be explained by the need to limit the scope seen by each peer. Whereas in dense urban spaces, the Chord structure is utilised to offer efficient routing and mediation effort, a scope of a range query will hardly include far away zones covered by the ring.

For this purpose, the DHT layer itself needs to be adapted to support the query needs. The latter can be summarised as follows: (i) queries with a small geographic scope and in dense urban subspaces are very frequent (high arrival rate) and (ii) the number of users is much higher than in a rural space. Therefore, a well meshed overlay and efficient query partition is needed, also limiting the scope of each peer to a small subspace leads to specialisation of nodes, and limits the need for remote queries (with several hops).

For less densely populated subspaces (e.g., rural area), the peers manage a larger geographic zone, since the number of users and their query frequency is much lower than in an urban dense space. The nodes assigned to this space do not have to be

well meshed with other urban nodes or far away nodes, but still have to be aware of predecessor and successor nodes besides the remote fingers.

Several aspects of the analytic model could be used in the design phase of a hypothetic system. The compromise between the accuracy of geographic data is, for instance, a major reason for increased complexity in the management overlay for a decentralised LBS. The way to compensate this is to take advantage of the fractal nature of the space filling curve used in addressing. Therefore, the partition of subspaces among overlay nodes applies a prefix partition of dense subspaces. The space description accuracy should arise from the density of the subspace. The density of subspaces arises from the number of objects and GIS systems pushing content in the DHT layer existing in a bounded subspace. This density is matched by the number of overlay nodes that are assigned to that subspace.

The objects offered by a GIS are addressed locally, therefore a spacial specialisation already emerges locally. The bootstrapping stage allows the content of the GIS servers to be placed along the Chord ring. Based on the number of nodes that discover each other due to predecessor and successor rules, it could be said that some clusters with closely stacked data items will emerge. These clusters are almost continuous, since they are allocating objects to keys which are separated by small or no gaps in the data Hilbert space. Based on this, some further optimisation of constructing the overlay could be achieved. This fractal property could be applied to tree-structure node scope and connectivity as follows:

- Geographic urban groups require well meshed peers, whereas far away nodes are less important in a finger table.
- Transition between large geographic zones need to be supported, to allow ubiquity of the system and support moving between cities or in rural subspaces.

Figure 4.13, demonstrates the addressing principles and scope of each node in the overlay. The construction procedure of the overlay relies on the granularity of the object IDs which are managed. The bootstrapping process is important in changing the structure of the DHT layer. It first starts by nodes discovering each other similar to the way it is defined in Chord. Now assuming that enough neighbouring peers already know about each other, the following optimisation can take place:

1. Each node identifies the minimum and maximum Hilbert IDs they manage and can select an ID representing the longest prefix match of the object IDs which are 2γ-bit long, representing the range $[ID_{min}, ID_{max}]$.

2. Each node receives an advertisement from its successor node advertising the range of keys it manages $[ID_{min}, ID_{max}]$. If none is received, the node decides to advertise its own known range to its predecessor.

3. A node which receives the $[ID_{min}, ID_{max}]$ range managed by its successor checks the density of the range and the gap between the received ID_{min} and its own ID_{max}.

4. If the gap is too large, then the advertisement is ignored and the information about the successor node is not passed to the next predecessor, and the successor node is informed.

5. If otherwise, the node adds $[ID_{min}, ID_{max}]$ to its own range swapping its own ID_{max} for the received ID_{max} from its successor, and adding its own ID_{min} to the range. The new range $[ID_{min}, ID_{max}]$ is passed further to its predecessor.

6. When a node receives a failed advertisement back from the last contacted successor node, then the cluster gathering a high density of objects is assumed to reach an end, and a new structuring takes place.

7. Based on the number of peers discovered between the gathered $[ID_{min}, ID_{max}]$ a prefix can be formed as the longest prefix match between the IDs in $[ID_{min}, ID_{max}]$. Each node is assigned a load to manage, which equals the

$$\frac{number\ of\ keys\ in\ the\ range\ [ID_{min}, ID_{max}]}{number\ of\ peers\ in\ the\ cluster}$$

.

8. Those nodes managing keys between $[ID_{min}, ID_{max}]$ form a Chord ring built according to the Chord protocol.

9. Nodes at the start and the end of the managed range join a higher hierarchy, to form another Chord ring at a different prefix level.

10. Nodes construct a higher level Chord ring similar to objects by advertising the IDs of Chord nodes they represent. A higher hierarchy is established as soon large gaps are again discovered in the ring.

The efficiency of the multiple Chord hierarchy is proven in the following section. In fact, a fully meshed Chord ring is used while the scope of the queries is analysed to find out how many nodes are actually involved in the query.

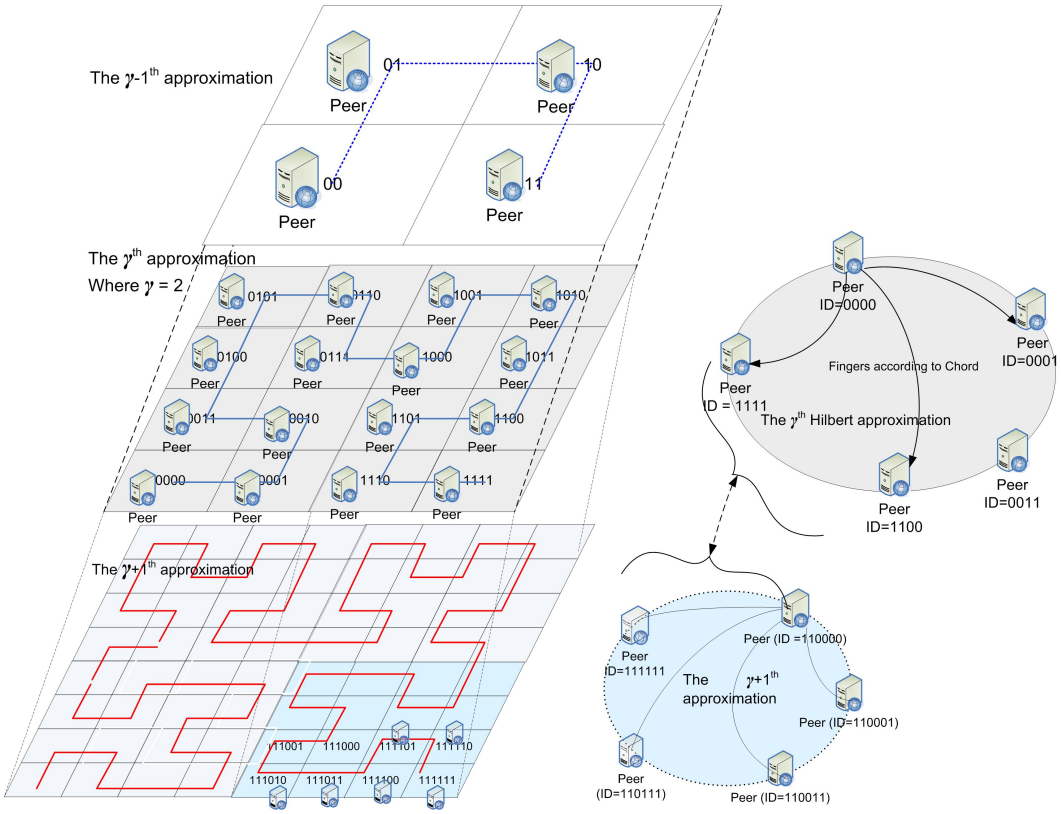

Figure 4.13: Hierarchical partition of peer IDs - Left: Hierarchical partition of peer IDs following the Hilbert 1^{st}, 2^{nd}, and 3^{rd} approximations; Right: Hierarchical approach to partitioning of geographically dense spaces to more granular Chord rings following the Hilbert 1^{st}, 2^{nd}, and 3^{rd} approximation [90]

4.6 Simulation Validation of Analytic Results

Based on the design criteria used in the analytic discussion, some elements of space modelling need to further be investigated. The way the query behaves and how the sys-

tem scales, can only be evaluated when using simulation. The evaluation methodology is aimed at:

1. Modelling the query needs (i.e., the number of queries per user movement). For this, a mobility model is needed with assumptions on user prior knowledge of movement and its direction.

2. Object density and its distribution among the peers is also modelled to mimic mostly shorter range wireless networks with overlapping covering. For this the aggregation methods of the address space and their effect on the query size are analysed.

3. The number of hops and the scope of the query is modelled to be dependent on knowledge of movement vector (movement direction and velocity) within an urban environment. This analysis should confirm the need of limiting the degree of each overlay node and support the need for a hierarchy.

4. Confirm the analytic estimate of query diameter and the number of messages involved in a query.

The effects of the optimisation rules designed for efficient overlay communication are also investigated and proven. First, the simulation models are presented.

4.6.1 Simulation Model

The simulation model is separated as *(i) network model*, *(ii) node model*, *(iii) information model*, and *(iv) user mobility model.*

4.6.1.1 Network Model

The underlying simulated network assumes a packet-switched IP layer, which, for the purpose of this work, has been implemented in a simplistic manner (details of the implementation have been given by Stenzer [172]). The network layer offers a fully reliable message forwarding service using IP addressing. The IP-layer is kept simple to allow for scalable implementation, which can support large overlay network topologies. The implemented overlay protocol is based on the Chord DHT protocol. An overlay network structure is constructed on top of the simulated reliable IP-layer. Bootstrapping

and maintenance of overlay routes rely on the Chord protocol. This simplified view of underlay networks is chosen, since it is sufficient to study the communication overhead and behaviour at the overlay level. This allows a more flexible simulation model which can support a large number of nodes.

4.6.1.2 Node Model

Most of the proposed functionality such as the addressing scheme, the clustering relies on locally implemented rules which when using the Chord overlay network, lead to the emergence of the semantic overlay. Nodes are simulated failure free, since the maintenance overhead of the overlay is out of the scope of this simulation study. In contrast, other dynamics have been taken into account, such as the validity of network context using soft states. The selected granularity of the grid mapping covering the earth surface requires a Hilbert approximation $k+n = 15$ requiring a $30-bit$ maximum address length in Chord for each cell. Assuming that since the granularity of $1m^2$ of an area covering about the surface of Germany requires 2^{38} cells, the above value of $k+n = m = 30$ represents an aggregation of each 2^8 cells together and assigning them to a single Chord key.

This value is taken as a compromise on the granularity chosen to describe geographic content. Each Chord key can be mapped to a unit subsquare of size $2^4 \times 2^4 m^2$ representing a bit change in the Chord ring. The Chord ring is constructed as a fully meshed Chord ring, i.e. every single peer is connected to $O(log(N_p))$ other peers. A portion of the Chord ring is, however, implemented to include several nodes responsible for a high concentration of objects. The initial configuration is done this way: $N_p = 2^{10+p}$, where N_p refers to the total number of peers in the ring, whereas $N'_p = 2^p$ refers to the number of nodes that cover a simulated geographic zone equivalent to an urban environment. The N'_p nodes are distributed geographically in a simulated urban area of the size $268km^2$. The urban space is simulated in a portion of a $300,000km^2$ area (symbolising a that of a country like Germany), which consists a continuous Hilbert cluster starting at ID_{min} and finishing at ID_{max}. The remaining $N_p - N'_p = 2^{10}$ nodes are distributed at logical equidistance along the Chord ring. A summary of the modelled space and matching Chord ring is provided in Table 4.3.

Parameter	Description
Network model	Chord nodes are simulated at the overlay level, i.e., fully reliable nodes and no packetisation nor delays
Overlay nodes	Both access and search peers are simulated. Chord is implemented. Node IDs are obtained using the Hilbert curve partition of the key space
Simulated object space	Using a fixed grid cell size of $16 \times 16m^2$, 30-bit key space ($\gamma = k + n = 15$)
Urban area	$268km^2$ large, where access networks and users are located, e.g., Passau in Germany is about $69km^2$ large
Peer distribution	2^{10+p} of which, 2^p responsible of the $268km^2$ city and the remaining 2^{10} for the rest of the earth

Table 4.3: Network and Overlay Model Parameters

4.6.1.3 Information Model

Given the geographic area of a medium-sized city, heterogeneous wireless cells are modelled as randomly distributed cells. Similar to [198], 50 wireless cells are distributed randomly for each $1km^2$. That makes about 13400 access networks for the whole $268km^2$ modelled city (See Table 4.3). Since wireless cells are heterogeneous in nature, different cell sizes are considered. Geographically speaking, a cell size is generated using the formula:

$$(16m \times 16m) * (2^s \times 2^s)$$

where $s \in \{1,\ 2,\ 3,\ 4\}$. For instance, cells whose radius is up to $128m$ can be positioned inside a square of area

$$256m \times 256m \equiv 16m \cdot 2^4 \times 16m \cdot 2^4$$

i.e., $s = 4$, where 2^{2s} represents the number of keys required to encode a given cell in the Chord space. The index s is varied uniformly between 1 and 4, when generating heterogeneous cells. A geographic granularity of about $16m$ can be described with these parameters. Analogous to the geographic information, mobility is also modelled to capture the user behaviour and therefore the way the queries need to be generated.

4.6.1.4 User Mobility Model

A graphical illustration of part of the urban city is shown in Figure 4.14. Further to the object model, the query is also modelled according to some movement pattern assumptions.

- For a single user's movement, a starting point is chosen, from which the user progresses along a Manhattan street system (similar to the one used in [36]). The user follows a random walk street model composed of a grid street system with junctions each $256m$, where the user randomly selects the next direction with probabilities 0.5 for continuing straight on and probability 0.25 for turning either right or left. For the city size of $268km^2$, this makes 64 horizontal streets and 64 vertical streets.

- For each scenario the user is assumed to have travelled a distance to 4080m, before the simulation reaches an end. The movement direction can change by each street junction, i.e., every $(\frac{256m}{velocity})$ seconds. This distance provides enough street portions positioned along different segments of the Hilbert curve. As a result, clusters can be found in a random way inside the search boxes which are placed along the user's movement path.

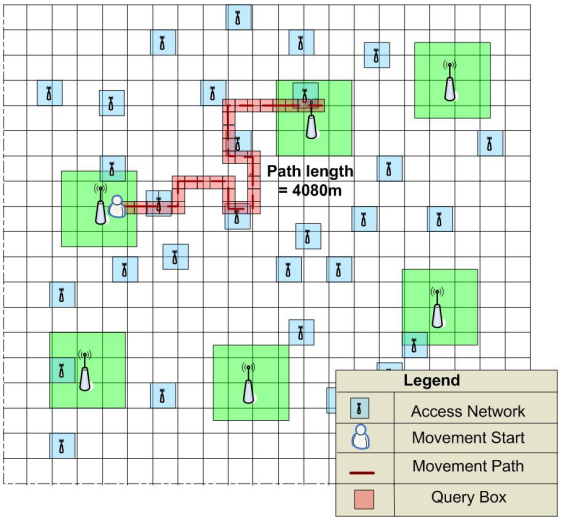

Figure 4.14: A portion of the simulated environment with example query boxes - Brownian motion through Manhattan street model

The query boxes are generated separately ahead of the an overlay simulation run, and could be said to have the following properties:

- Assuming smaller query boxes as the full length of the traversed path, query boxes are emitted by the user recursively as he/she progresses along the movement path.
- A query box cannot be generated as long as the user has not moved out the area where his previous query has be generated, this prevents slow users from generating large queries too frequently.
- A user moving with a given linear velocity v_{lin} crosses each single Chord key every 16m (i.e., every $\frac{16}{v_{lin}}$ sec). In this special case, and due to the Manhattan street model, movement angles do not have to be taken into account.
- Each side of a square query box covers a distance of $2^k \times 16m$ requiring a new query box every $\Delta\tau = \frac{2^k \times 16m}{v_{lin}}$ sec.
- Given a certain movement velocity v_{lin} and a sampling interval $\Delta\tau$, the minimum query size could be formulated as $k_{min} = \log_2(\frac{\Delta\tau * v_{lin}}{16m})$, where k integer and $k \geq 0$. The $16m$ distance refers to the minimum Euclidean distance representing a adding or subtracting a single bit in the Chord key space.
- The size of the search box that covers a whole street portion (i.e., from junction to junction) is $k_{street} = \log_2(\frac{256m}{16m}) = 4$.
- The value of k is bounded in all scenarios between $k_{min} \leq k \leq k_{max}$, where $k_{max} = 6$.
- To cover the full distance of about $4km$ movement path, for each k the query has to be repeated Q times, where $Q_{max} = \frac{4080m}{2^k \times 16m}$ (plotted in Figure 4.20).

The messages sent back as part of a response are also measured as both message units (where each unit represents an additional Chord key sent back). More formally, the query process is dependent on the velocity of the movement:

Movement dependent query definition Assume a MN travelling with a linear velocity v_{lin} in $metres/sec$ and each side of a query box covers a distance of $2k * \Delta d$ in m, where Δd is the minimum distance change between two neighbouring Chord keys

when adding or subtracting a single bit change. Given that the user generates queries periodically with a sampling interval $\Delta\tau$, then the query box could be defined as follows:

$$k_{min} = \left\lceil \log_2 \frac{v_{lin} * \Delta\tau}{\Delta d} \right\rceil$$

The minimum search window size k_{min} is needed when talking about slow moving users. Assuming three different speeds ($2m/sec$, $8m/sec$, and $16m/sec$) corresponding to ($7.2km/h$, $28.8km/h$, and $57.6km/h$) respectively, and different periods for generating queries $\Delta\tau = 1,\ 2,\ 4,\ 8,\ 16,\ 32,\ 64\ sec$, the resulting minimum query size is given in Table 4.4.

$v[m/s]$	$\Delta\tau = 1$	$\Delta\tau = 2$	$\Delta\tau = 4$	$\Delta\tau = 8$	$\Delta\tau = 16$	$\Delta\tau = 32$	$\Delta\tau = 64$
2	/	/	/	0	1	2	3
8	/	0	1	2	3	4	5
16	0	1	2	3	4	5	6

Table 4.4: Minimum query box size k_{min} depending on velocity v_{lin} vs. sampling interval $\Delta\tau$ (in seconds)

Search Complexity Along a Manhattan Path Given the grid mapping of a given finite square space which has the size $2^{k+n} \times 2^{k+n} cells$ (where the smallest cell is $2^f m \times 2^f m$ large). Furthermore, for a Manhattan path of length $(2^l - 2^{l'})m$ and a query box of size $2^k \times 2^k$ used to query the along the Manhattan path. And given that N_p the number of nodes forming part of the Chord ring. l, l', f and f' are integer indices.

Then, the number of search boxes required to cover the length of a straight path is given as Q_{max},where:

$$Q_{max} = \left\lceil \frac{(2^l - 2^{l'})}{2^f * 2^k} \right\rceil$$

While referring to the query overhead per search box given in the formula in Definition 4.4.2, where M is dependent on both the granularity of the object addressing and their partition among the peers:

$$O(N_2(k, k+n) * (log(N_p) + M))$$

Now the total query effort along a straight Manhattan path can be given as:

$$O(Q * N_2(k, k+n) * (log(N_p) + M)) \tag{4.2}$$

It must be said however that since $k_{max} = 6$ also means that several Manhattan blocks exist within such a large query. If the movement generation randomly selects turns after each street junction, a single $2^6 \times 2^6$ query box might be sufficient to cover the whole 4km long path. For this reason, the selected path is first left fixed, while varying the query box. If a given portion of the Manhattan street model has been queried, the query only targets the new clusters that have not been addressed before.

4.7 Simulation Analysis

4.7.1 Query Overhead vs. Query Results

Without taking message aggregation into account, it is first assumed that a search occurs for every single cluster at the data level. The query is therefore defined at the Hilbert curve level representing $\gamma = 30$. The size of the query box is gradually increased for each scenario from $k = 0$ till $k = 6$, where $k = 0$ represents a search box of geographic size $16m \times 16m$. Since the Manhattan street system fits with the grid system used to model the geographic subspace, the query box defined by $k = 0$ also represents querying a single key in the Chord key space along the movement path of the user. This is equivalent to $get(key)$ query in Chord (instead of $get(range)$), and should incur into $O(log(N_p))$ messages. This is then repeated $Q_{k=0} = \frac{4080}{16} = 255$. The number of messages which should be shown in the graph is bounded by $O(Q_{k=0} * log(N_p)) = 5100$.

Since the $get(key)$ effort might vary, the result shown in Figure 4.15 for $k = 0$ is the simulated bound measured over several runs.

The involved overlay messages including all the messages involved to cover each query box is measured. For $k > 0$ several Hilbert clusters might exist (recall $N_2(k, k+n)$ bounded by 2^k clusters). For each cluster a $get(range)$ takes place. For the same path length, however with different query box sizes, all involved *search message* units shown in Figure 4.15. In addition to all routing messages involved in the search, Figure 4.15 also illustrates the number of response message units, which are sent back by each peer to the originator of the search in the simulation. The responses are not filtered. They only represent each object sent for each key queried in the Chord ring.

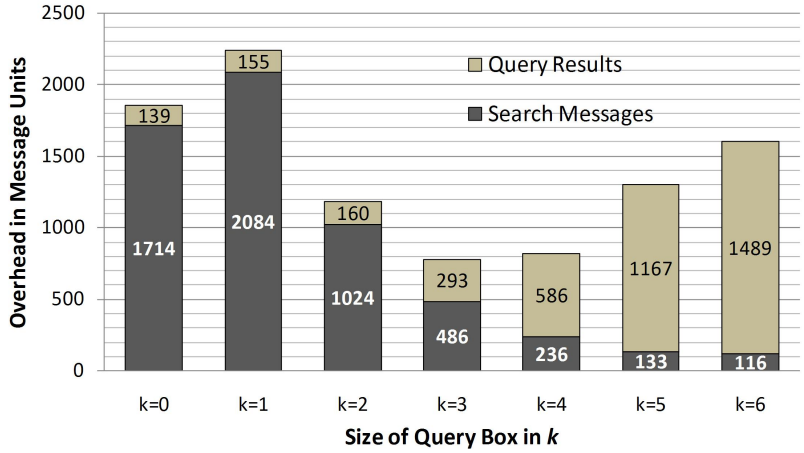

Figure 4.15: Communication overhead in message units - number search and response message units

Since objects are distributed in a rather random way, for the 4080m path, the number of responses would vary depending on the cells whose coverage will overlap geometrically with the scope of the search box. For each overlapping cell (of a $16m \times 16m$ surface), an object encoded with a single Chord key is found, generating a response. Aggregating responses so that each peer will only send a single message is not plotted in Figure 4.15.

When summing up the message overhead for both the search and response effort, the ideal size of the query box is found to be $k = 3$ or $k = 4$. The number of query messages decreases with increasing k, whereas the bulk response information size increases (for the whole path). For $k = 0$, the shown results in Figure 4.15 are those of an exact match query occurring along the movement path, and therefore, repeating $Q = 255$ times (see plotted Q in Figure 4.20). The first range query occurs by $k = 1$ resulting in a large amount of messages. Theoretically the multiplier $Q * N_2(k, k+n)$ stays the same for all query boxes, however, in the simulated earth partition, the number of clusters within each query box is found to be a lot smaller than the theoretic asymptotic limit $N_2(k, k + n)$ (see Figure 4.16). This results in a decreasing trend of the total query messages in Figure 4.15.

4.7.2 Examining the Number of Hilbert Clusters

To examine the number of the clusters along other portions of the simulated urban space, the path generator and corresponding Hilbert transformation is carried out on a

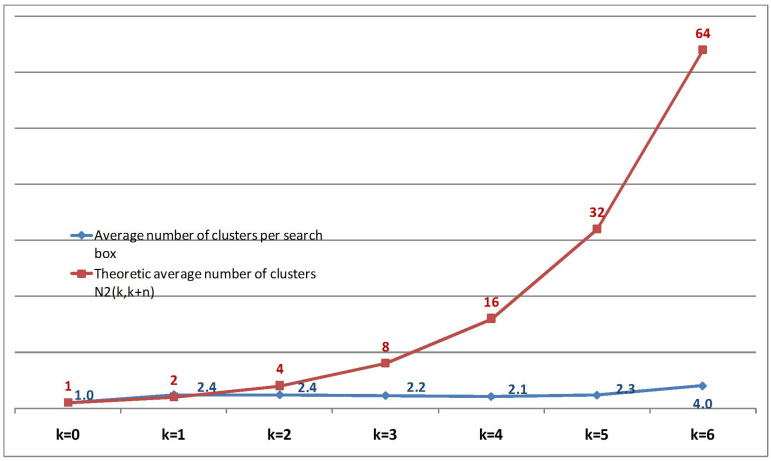

Figure 4.16: Number of theoretic number of clusters v.s. found clusters in simulation scenario per search box - The clusters defined in the simulation show the pessimistic theoretical average with approaches 2^k for a query box of size k

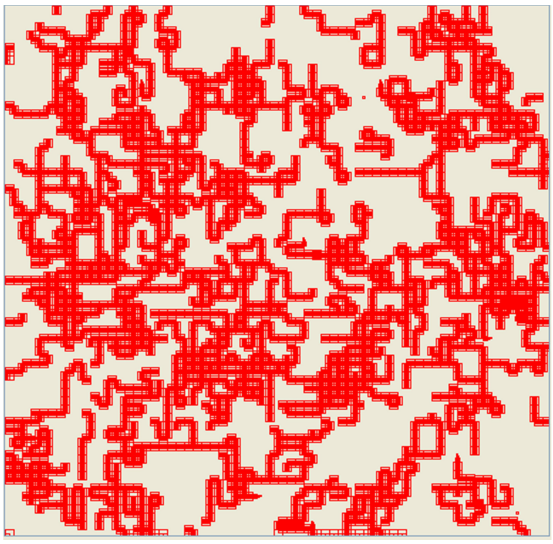

Figure 4.17: Modelled urban environment - Randomly placing 200 query paths in the urban environment; each path is 4080 m long

large scale to test the accuracy of the simulation model.

First, 200 paths of length 4080m are placed randomly across the simulated area (an overview of the modelled urban area is shown in Figure 4.17). For each path, 4 scenarios, in which k is increased from 1 to 4, are investigated. For the same path the number of query boxes Q varies. The simulation, then, returns the number of Hilbert clusters found in each query box.

Figure 4.18 shows the distribution of number of clusters found per query box. On the x-axis the number of found clusters per search box is given, while the y-axis represents number of cases per setup, where each setup is repeated per search box size k. The total number of search boxes is given in the title box for each histogram. With smaller k the number of queried boxes Q is higher than with larger values of k. However, it turns out that the greatest number of cases shows that the number of clusters is much lower than the theoretic value $N_2(k, k+n)$, which is bounded by 2^k. The maximum number of clusters is in most cases close to 2^k, but occurs quite rarely.

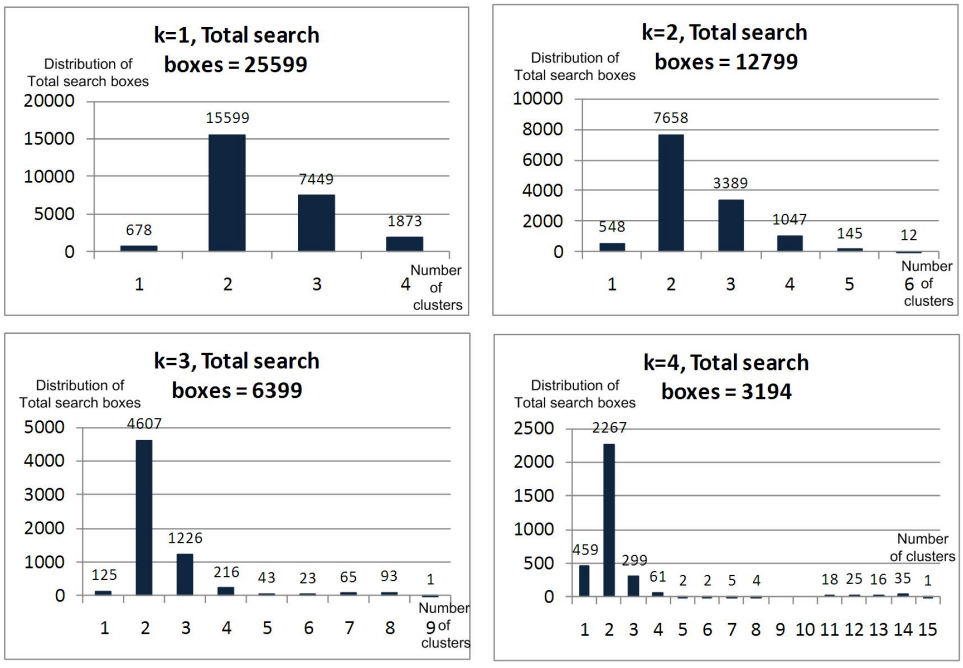

Figure 4.18: Simulative number of cluster for search boxes of size (for $k = 1, 2, 3, 4$) placed randomly in the modelled city - Distribution of number of clusters found in 200 selected 4080m paths per each search box size

4.7.3 Query Response and Information Error

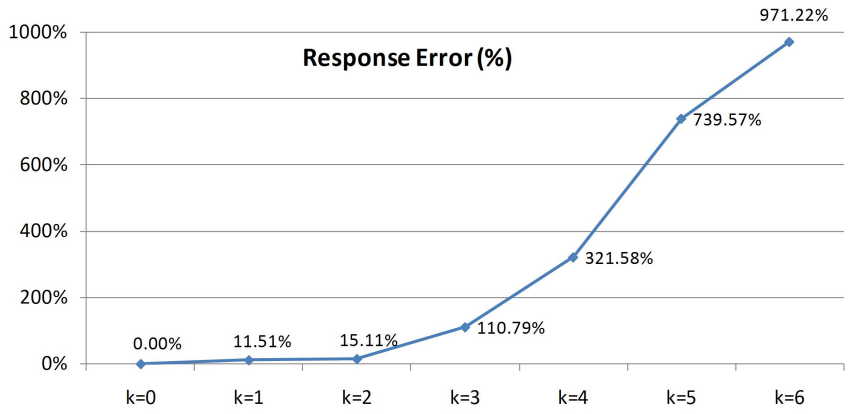

Figure 4.19: Query Response Error - Given in percent (%)
$Response\ Error = \frac{Number\ of\ Responses\ for k-\ Number\ of\ responses\ for\ k=0}{Number\ of\ responses\ for\ k=0}$

According to the results shown in Figure 4.15, in order to reduce the number of search messages, the size of the search box can be increased. This limits the numbers of search boxes needed to query the same travelled distance. However, the resulting response messages increase as a result. Without further filtering at the queried nodes, the response messages carry an error margin. In the targeted case study, only those access networks, whose coverage is geographically close enough to the movement path of the user, are of interest (in order to predict the handover). In other words, the targeted wireless access points are those that are covered by the smallest search box possible along the movement path (i.e., $k = 0$). The responses to this query could be said to be the ideal responses fulfilling the user's query needs. The larger search boxes can result in larger numbers of clusters, but according to the results in Figure 4.18 this happens only rarely. The response messages measured in Figure 4.15 increase for increasing the size of the search box (i.e., $k > 0$). This behaviour is captured as a *"semantic"* query error, which is estimated for each search box size k as follows:

$$Query\_Error(k) = \frac{Number\ of\ Responses\ for\ k-\ Number\ of\ responses\ for\ k=0}{Number\ of\ responses\ for\ k=0}. \tag{4.3}$$

The query error is plotted in Figure 4.19. For the ideal query size $k = 3$ and $k = 4$, the query error approaches vary between 111% and 322%. The query error only indicates a computational cost, since the responding nodes or even the search peer have to filter out the messages that are sent back. In some query types such as point of interest queries, these error results are actually a desired aspect (e.g., search for any WLAN access points within a $800m$ radius).

4.7.4 Overlay Behaviour

The structure and partition of content among the peers, affects the number of response messages really needed to encapsulate the responses to a range query. The results shown in Figure 4.15 plot the number of objects associated per each key queried, which produces a response. In a more practical implementation, the response messages have to be aggregated or filtered at the access peer level.

4.7.4.1 Response Message Aggregation

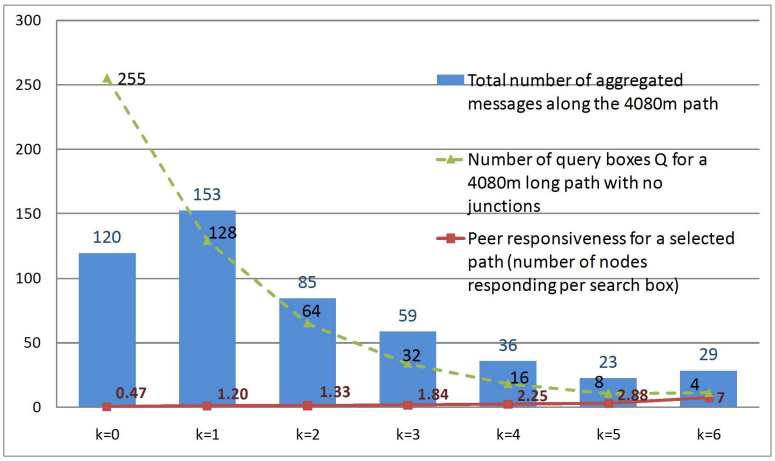

Figure 4.20: Number of responding peers per search box - Aggregated responses representing each time a peer responds to a separate query, where each query is sent for each cluster

Once a peer receives a query message (covering a cluster length), the query can be further processed in a similar way to SQL high level query definition. Instead, of looking at the complexity of data base functionality, this work is only centred on the networking and pure communication overhead. Therefore, the simplest technique to

reduce the communication overhead is to aggregate the message units shown in Figure 4.15 in a single variable sized message. Each time peer is queried, a response message is generated gathering all possible objects associated with the queried cluster. The resulting messages for the whole simulated path of $4080m$ is illustrated in Figure 4.20.

For $k = 0$, the number of aggregated response messages is 120, when compared with the number of search boxes 255, only 47% of search boxes have led to a response. This result can be interpreted at the number of peers responding per query box is for the whole modelled path averaged to about 0.5 peers per search box.

For $k = 1$, the first range query takes place requiring each search box to be split into $N_2(k, k + n)$ clusters. Each time a cluster is queried, the number of peers that send an aggregated response is measured. For $k = 1$, 128 search boxes result for the whole movement path are defined, of which 153 peers send a response (due to splitting a search per cluster). This makes the average number of responding peers per search box by 1.2.

For $k > 4$, the possibility of a search box to cover more than one street portion increases. The number of required query boxes Q will vary according to the starting point and to whether the user turns after the first junction. This is due to the geometry of the Manhattan street system which has a junction each $256m$, whereas, the side length of a query box of size $k = 5$ is $2^5 \times 16m = 512m$. If the path turns at the first junction, part of the next street section has already been covered by the previous query. Only those clusters which have not been queried are searched. The only exception is when $Q = Q_{max}$, which is plotted in Figure 4.20 for $k = 5$ and $k = 6$.

4.7.4.2 Scope of Chord Ring

Search box size	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
Simulative $Q * N_2(k, k + n)$	255	307	153	71	35	18	16

Table 4.5: Measured simulated number of clusters (ranges) found on a randomly chosen single path of length $4080m$ equivalent to $Q * N_2(k, k + n)$

The scope of the Chord ring refers to the number of overlay hops actually involved in a box query. This is theoretically bounded by the $O(Log(N_p))$ cost defined in Chord

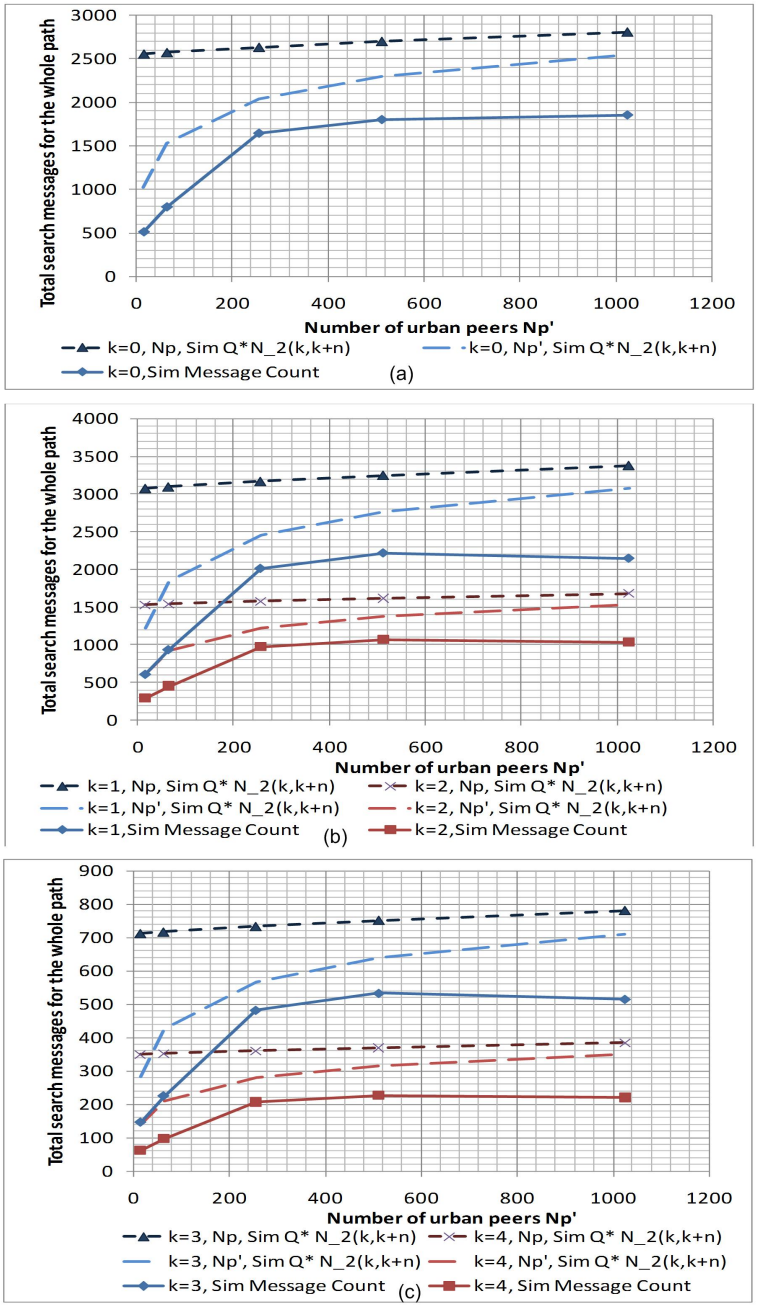

Figure 4.21: Numerical vs. simulative number of search messages for the whole 4080m path, when varying the number of peers per urban area N'_p - Message overhead $Q * N_2(k, k+n) * log(N_{peers})$ for *(i)* $N_{peer} = N_p$ and simulative number of clusters for the whole path ($Sim\ Q * N_2(k+k+n)$), *(ii)* $N_{peer} = N'_p$ and simulative number of clusters for the whole path ($Sim\ Q * N_2(k+k+n)$), *(iii)* measured simulation total overhead; *(a)* $k = 0$ for pure $get(key)$ case; *(b)* comparing $k = 1$ with $k = 2$; *(c)* comparing $k = 3$ with $k = 4$

[175]. However, as already discussed analytically, the Hilbert ID space is mapped one-to-one to the Chord ID space, removing the need for a fully meshed Chord ring. In fact, the mobile queries are expected to involve a much smaller number of nodes than the theoretic bound of $O(log(N_p))$, due to the nature of the query in mobile scenarios, where the queried nodes are not that far away from the origin of the query (in terms of ID space). This requirement has resulted in the hierarchical structuring of the global overlay network, with Chord being used to construct a ring structure at each hierarchy level.

In order to prove the above hypothesis relating to the scope of the query and the required hierarchy, simulation is used. The simulation assumes a fully meshed Chord ring, where its participating N_p are assigned a much larger ID space, mapped to a relatively large geographic space. The query, however, is initiated inside a much smaller geographic scope (i.e., inside the urban subspace only), which is managed by N'_p nodes, where $N'_p << N_p$. Independent of the size of the query box, the number of involved hops should be bounded by $O(Log(N'_p))$ rather than $O(Log(N_p))$.

For this test, the number of nodes in the urban subspace is varied, while the number of nodes assigned to the rest of the modelled geographic space stays the same (i.e., $N_p - N'_p = 1024$). The number of the urban peers, indicated in the Table 4.3 parameter $p = 4,\ 6,\ 8,\ 9,\ 10$ corresponds to $N'_p = 16,\ 64,\ 256,\ 512,\ 1024$ nodes respectively.

For a single fixed path (to avoid the variability of total number of clusters) is queried, at each time a different number of urban peers N'_p. The number of found clusters is independent of the number of nodes, and is given in Table 4.5. This number is equivalent to the simulation estimate of the number of clusters (which is equivalent to the numeric result $Q * N_2(k, k+n)$). This value depends on the size of the query box defined through the parameter k, and is shown in Figure 4.18 to vary from one path to the next.

The number of query messages is bounded by the numerical result given in Equation 4.2 as:

$$O(Q * N_2(k, k+n) * (log(N_{peer}) + M))$$

This equation is used to compare the different hypotheses, by *(i)* fixing the multiplier part $Q * N_2(k, k+n)$ to the measured value listed for each k in Table 4.5; *(ii)* exploring the logarithmic routing cost due to the scope of the Chord ring, by testing $N_{peer} = N_p$

then $N_{peer} = N'_p$. The resulting graphs are compared with *(iii)* the purely simulated number of query messages gathered for each simulation setup.

The resulting graphs in Figure 4.21 can be grouped for each k, with three lines comparing the numerical estimate of the query overhead for the full Manhattan path when using the two values of N_{peer} (N_p vs. N'_p), with the actual simulation result.

Three cases are illustrated together *(a)* $k = 0$, *(b)* $k = 1$ vs. $k = 2$, and *(c)* $k = 3$ vs. $k = 4$. For $k = 0$, there are no clusters as already discussed but rather exact query retrieval of single keys. This means that the multiplying parameter $Q * N_2(k, k + n)$ is equal to Q, which is 255 queries for the whole path. The resulting measured overhead increases in a logarithmic way for increasing N'_p. The same tendency is noted when taking the numeric result with $N_{peer} = N'_p$. This is not the case when using the value $N_{peer} = N_p = 1024 + N'_p$. in the numeric equation. The difference between the two lines (numeric N'_p and simulation result), is a result of the difference between the actual number of hops needed by each query box in the simulation, which turns out to be below the theoretic upper bound of $O(log(N'_p))$.

The same tendency can be said for all the other cases of k plotted in Figure 4.21. In some cases, the simulation result decreases for $N'_p = 1024$, when compared with $N'_p = 512$, because the number of hops in both cases are similar (while exponentially distributed), and the simulation in the $N'_p = 1024$ had at least one hop less than the $N'_p = 512$ case.

The distribution of the number of hops per simulation run is shown in Figure 4.22. The number of hops required to reach a cluster (Chord hops) for $k = 3$ results in the histogram shown in Figure 4.22. The trials are repeated for the original setup with $N'_p = 1024$ nodes, for several hundred of times. For $k = 3$, the number of clusters along the full path is *simulative* $Q * N_2(k, k + n)$ varies according to the distribution shown in Figure 4.18. The query is repeated over different paths, where the number of hops to reach the head of each cluster (theoretically bounded by $O(log(N_{peer}))$) is recorded. The number of Chord hops needed to reach each of the resulting 1577 clusters are returned.

The resulting number of hops is plotted in Figure 4.22. The distribution of the number of hops is shown in the histogram along the cumulative distribution function. In over 90% of the cases the number of hops is less than 8 which is less than $log(N'_p)$,

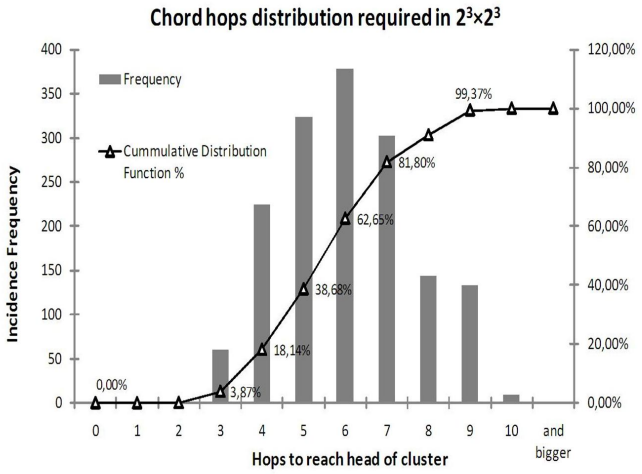

Figure 4.22: Number of Chord Hops averaged over 100 trials - Number of $get(range)$ calls is 1577

where $N'_p = 1024$. The worse case occurs in a few cases, where the number of hops reaches 10 (i.e., $log(N'_p)$, for $N'_p = 1024$).

The results confirm the hypothesis that the number of peers involved in a range query in an urban subspace is bounded by a logarithmic cost equivalent to a that of a Chord ring, which includes only those nodes in the urban subspace. Therefore, adding hierarchy as explained towards the end of Section 4.3 is necessary. This has the effect on the size of the routing table a node has to keep, which is then reduced to $log(N'_p)$ instead of $log(N_p)$.

As a result, the maintenance cost for the Chord structure itself, which involves a single update process of each entry in a routing table, is also considerably reduced.

4.7.4.3 Scalability and Resilience to Churn

The designed system could be said to be capable of supporting large numbers of users due to the partitioning of the system into two interacting overlays; the overlay of search peers and the overlay of access peers. The assignment of mobile users to search peers could be similar to assigning users to proxy home agents in mobile IP with a limitted geographic scope. Each proxy home agent can be dimensioned to support a large number of users. This approach is similar to the ROAM system, which has been thoroughly compared with mobile IP in [201].

The overlay of access peers first interacts with the search peers at bootstrapping, where a large amount of data traffic can be generated to provide the search peers an initial view of their geographic scope. Following an early heavy transient phase, a steady state is reached where search peers are updated with regular changes to the wireless resources. This update process occurs for each wireless system within the scope of an access peer and is its arrival time by each access peer is exponentially distributed. The updates are reported to the search peers through a publish/subscribe mechanism.

In the simulation study, the study of the Hilbert clusters randomly placed on the whole surface of the modelled urban space also models the complexity introduced by having a large number of search peers. The effect on the access overlay has been thoroughly studied.

The search peer has to be able to respond to large number of localised queries at a frequency similar to HLR or VLR databases in cellular networks. The arrival rate of mobile users at a search peer could be modelled as $\lambda_{arr}(sec^{-1})$. This arrival rate depends mostly on the density of the users in a given geographic area, and less on the speed at which the users are moving. Since higher velocity only means shorter management sessions at a given search peer, however, larger query scopes and, therefore, a higher filtering process.

A cache at the search peer would facilitate responding to high velocity users higher real-time requirements, as the information can be generated from local the local cache, rather than be queried in the access peers overlay. In the case of the existence of a cache at the search peer, queries are started for two purposes:

- Updating the cache within a validity interval (representing a time interval in which the stored spatio-temporal results of the query are assumed to be valid). Each search peer sends a request to access peers to obtain the most up-to-date view of the object subspace based on arrival rate of $\lambda_{update} = \frac{1}{validity\ time}(sec^{-1})$.

- Querying other subspaces not stored locally, which are defined by the query. Therefore, it is important to define the right aggregation parameters that assign users with similar behaviour (e.g., fast users vs. slow users) to the same search peer.

The design of an accurate caching strategy is beyond the scope of this work, but it could be influenced by different system requirements such as user behaviour prediction,

user behaviour aggregation, load balancing between access peers. Since queries are started in an asynchronous manner, the arrival rate of queries in part of the ring could be said to approach the frequency of updating the cache of each search peer times the number of local search peers attached within a given subspace.

$$\lambda_{update} * N_{search\ peers\ in\ subspace}$$

While the number of search peers required to guarantee a load-balanced operation of the whole system is hard to predict, the structure of access peers overlay could be shaped to offer load balancing. Given that access peers send information about the key range they manage, further information of the number of objects each node locally stores, can be sent during the bootstrapping process. Based on this information, the number of Chord keys managed by each node can be modified. The following heuristic could be used. The higher the number of locally stored objects, the larger the number of queries that need to be dealt with. Therefore, the larger the density of the locally stored objects, the smaller the assigned Chord key space should become. Whereas, nodes with lower object density can be assigned a larger Chord key range to manage. Similar issues are addressed in the literature on load-balancing DHT-based overlay networks (e.g. see [23]).

The overlay nodes taking part in the system could also be assumed to have a low churn, since they are physically management nodes designed to be able to cater for multiple and parallel requests simultaneously. To counter failure, however, some redundancy could be introduced in the system by encapsulating the mediation information (stored objects and their location) in a virtual machine, which is duplicated on several other redundant nodes.

4.8 Chapter Summary

This chapter develops a design methodology to construct semantic overlays capable of efficiently manage scattered context information among heterogeneous GIS systems. Space modelling is applied to the context data describing wireless access networks, and a geographic clustering is derived thanks to the Hilbert space filling curve.

With the help of the Hilbert curve, space is modelled and partitioned in a way, which allows spatial specialisation (i.e., a 2-dimensional space is partitioned in separate

zones representing data management domains, and later overlay clusters). A separate modelling of space has an important impact on the resulting system. Dense areas (e.g., urban areas) with large numbers of data items and GIS elements, would require a higher approximation of the fractal representation of space, whereas low density areas (e.g., rural areas) do not. This spatial specialisation is used to assign zones to peers and to structure the overlay graph itself.

The fractal nature of the curve defines a prefix-based addressing and organisation of the data space, which can be matched by the network construction of the semantic overlay. This structuring and its effects in mobile environments is tested and evaluated through analytic study of communication overhead and some mobility-based query modelling. To further adapt queries to user demands (or user context), simulation is used to implement a mobile scenario taking place in an urban environment. The simulation results demonstrate the viability of the design process, which can be studied analytically.

Given that a mobile user can retrieve the network context through a semantic P2P LBS, this process should replace the need to discover link technologies on lower layers during movement. Reducing beaconing should have the effect to reduce unnecessary network discovery effort on several network interfaces in parallel (which in turn costs energy). Context retrieval is, instead, carried out in the background, while connected to a single wireless network. Furthermore, the needs of the user can easily be specified or enriched at the search peer, which takes the role of a topology discovery proxy for the mobile user. In the next chapter, the effect of context on lower layers and the way to interact with them is explained. For this purpose a prototype based on two well-known protocols (namely WLAN and MIPv6) is built and analysed in a simulation framework.

Chapter 5

Case Study: Location-Based Handover Execution

In this chapter, the implementation of a context-aware handover is given. An important component of the context-aware handover is the execution of the handover procedure based resulting from the decision or selection algorithm as explained in Chapter 3. Context-aware handover is defined as the process of selecting the best wireless network that most suits the user behaviour, activity, movement, and other elements of the user context. To support this intelligent selection process, network context is used to describe the heterogeneous wireless domains in a standardised manner. Triggering a context-aware vertical handover between different link and networking technologies and standards remains a challenge. The interaction between handover algorithms (decision algorithms) and the context information describing networks is restricted by the programmability of the handover process itself. Network context has to include functional information, which indicates to a handover decision module how to trigger a handover (i.e. programme the handover procedure) for the given link technology and the network domain. This context information represents an ontological abstraction of the handover process for each wireless domain. In this chapter the way context information is extracted from a wireless network, and the way to interact with this information is demonstrated. For this purpose a focus is made on two wireless domains, which each consist of a single WLAN access point supporting the IEEE 802.11 standard [2] at the link layer (layer 2), and connected through Mobile IPv6 [98] at the network layer (layer 3). By looking at the details of the latter two protocol layers and their functionality, the

abstraction of handover information needed to describe the handover process as context information is demonstrated. A location-based handover decision algorithm is assumed to trigger the context-rich handover. A more sophisticated handover algorithm could be partly based on one of the alternatives discussed in Chapter 3 Section 3.2. The choice of technologies or standards is a matter of implementation. The two standards are well understood and have been researched thoroughly in the last few years (see Chapter 2). Developments on fast-handover in each of these standards are of great interest, however, these developments are incremental and therefore could still be included in the proposed context-aware architecture. The main hypothesis proven in this chapter is that when using any given link or network layer variation, context-aware handover can allow a better performance without major changes to the underlaying standards. This can only be achieved after a thorough understanding of the functionality of each protocol. Furthermore, what context-aware handover achieves is a universal framework to discover and take advantage of cognitive capabilities in the mobile devices, which are involved in the handover process. The way to interact with underlaying layers is shown through the scenario discussed in the remainder of the chapter.

5.1 Understanding IEEE 802.11's Mobility Management

In a first step, a precise description of the way IEEE 802.11 deals with mobility is conducted. This procedure applies to the original IEEE 802.11 family of standards *a*, *b*, and *g* [2]. The family above of standards will be referred to as WLAN in the rest of the chapter.

5.1.1 Handover in IEEE 802.11

5.1.1.1 Movement detection

Movement detection, as a process, defines the condition of the wireless link under which movement can be assumed. The mobile node (MN) assumes that it has moved after it has sensed a long term unreachability of the old AP. Detection of unreachability differs in whether it is a network- or a client-initiated handover process. In the case of a network-initiated handover, the signal strength measurements carried out by the node are sent back to the AP. When a certain threshold is under-run the AP sends a disassociation message to the MN. This method suffers from the drawback of the ping pong effect,

which leads to a drop of a MN due to temporary signal fading. Furthermore, the AP assumes that the MN can receive the disassociation message despite the bad quality of the signal due to movement. More common in WLAN is the MN-driven movement detection. In this case, the MN starts probing the old AP with probe requests that result into response messages. After a number of failed beacon responses from the old AP, other APs are searched for by sending a probe on the remaining frequency channels, known to the standard. The probing is also initiated to discover the reason of the initial failure. This could be due to movement, temporary fading of signal, or to collisions.

Another way to detect movement in wireless scenarios is a drop of the *signal to noise ratio (SNR)* beyond reparation. Most 802.11 cards implement a reduction of the bit rate and use a *request to send / clear to send (RTS/CTS)* handshake mechanisms to be able to reconnect to the old AP in case of temporary fading [2].

5.1.1.2 Search for a new AP

After all attempts to reassociate with the old AP have failed, the 802.11 standard defines two different search modes: passive and active scanning. In the passive mode, the MN only listens for beacon messages from surrounding APs. This method has a major drawback as the MN is unaware of the beacon period of the different APs. Therefore, if no AP is setup for a channel which is passively scanned by a MN, it is unclear to the MN when to stop waiting for beacons. In case of a standard beacon period of 100 ms, the MN would therefore have to wait more than a second to complete the search for a new AP since the search is carried out for the 11 to 13 WiFi channels before selecting an AP. It is worth noting that the number of channels depends on the frequency bands used by the IEEE 802.11 standard family.

On the other hand, in the active mode, the MN anticipates the search by sending probe broadcast frames on each channel. The APs, then, respond accordingly. The MN has to wait a given time called "$MinChannelTime$" for a response by APs on each probed channel.

In case of detecting other activity in the channel during the $MinChannelTime$, a second timer called $MaxChannelTime$ is used to cater for the delay of a response by an AP due to contention (collisions lead to retransmissions in 802.11). If the channel is sensed to be idle during a the time $MinChannelTime$, the channel is considered empty.

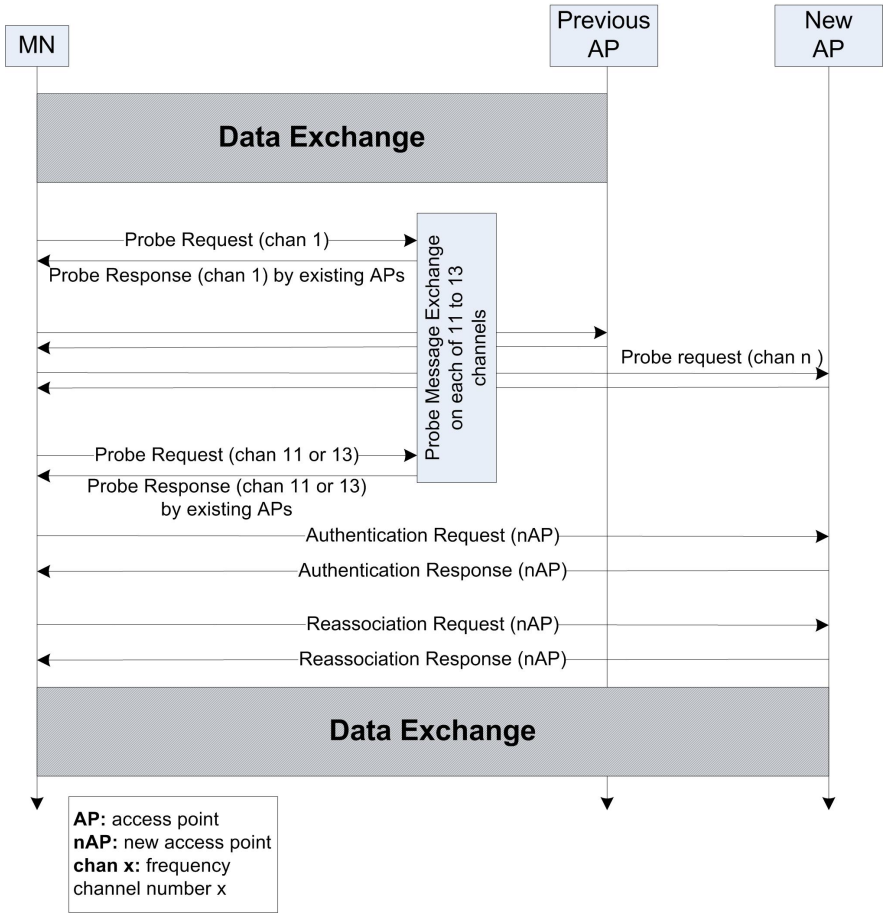

Figure 5.1: Handover progress between IEEE 802.11 access points - message sequence chart

This explains a performance degradation of the handover procedure when several MNs are attached to a single AP.

The total probing delay is given by:

$$ProbingDelay = MaxChannelTime(fortheusedchannel)$$
$$+ numberofemptychannels \times MinChannelTime \quad (5.1)$$

5.1.1.3 Execution

Once the list of APs resulting from the completed search known for all channels, the AP with the strongest signal can be selected. The execution of the handover is then done in two steps: *reassociation* and *authentication*. During reassociation, the MN sends a request to the new AP which in return sends an acknowledgment indicating successful association as shown in Figure 5.1. In the authentication phase, different types of encryption mechanisms will add to the delay.

5.1.2 Delay Studies of Protocol Interactions

Several wireless cards of different manufacturers are analysed in [185] by looking at the delay experienced by VoIP packets during handover. The measurements conducted show that the differences in implementing 802.11 have a large implication on the handover latency. The different cards might implement different variations of passive or active handover and might vary slightly in the movement detection procedure. A distinction is made between upstream and downstream delay due to the slight change in the procedure caused by the two different flows. The single-station scenario also showed a better latency compared to competing-traffic scenarios where delays grew considerably. Contention could cause some confusion since errors also occur due to increased interference rather than unreachability of the AP. The handover procedure took between (81ms, 89.5ms) mean measured delay-pair for (upstream, downstream) flows, in the case of Lucent-made cards, and (211ms, 213.9ms) mean delay-pair for the D-link-made cards. This delay is mostly accumulated during the search phase, which represents 90.8% and 90% of the total handover delay incurred by the two cards respectively (see probing phase in Figure 5.1). Both cases assume an active search by the MN rather than receiving advertisements from APs. The mean measured delay-pair in the case of the better

performing Lucent card increased to (181.3ms, 142ms), i.e., 189% increase, in busy networks that include other competing nodes. Solutions based on bi-directional streaming and buffering prove to be costly in the case of a frequent movement where new stations arrive and leave in an unpredictable fashion. One obvious solution to reducing the total delay is improving the search phase.

5.2 Understanding the Mobile IPv6 Protocol

Seen from the mobile node, a MIPv6 handover process relies purely on network layer messages. Differently to fast-handover proposals, no cross layering is assumed by the simplest handover procedure in MIPv6.

5.2.1 Handover Triggering in MIPv6

As already explained in Section 2.5, the *neighbor unreachability detection* (NUD) stands at the heart of handover management in MIPv6. Movement detection and handover execution at the mobile terminal are more of concern here. The binding delay and other efficient routing solutions have been addressed in Section 2.5.

5.2.1.1 Movement detection

There are two different approaches to movement detection in MIPv6. The first approach interprets failure to communicate with the old *access router* (AR) as being due to movement, while the second approach interprets a sudden detection of a new AR as being due to movement.

According to NUD in MIPv6, a *router interval option (RIO)* indicates at what frequency an AR will send *router advertisements (RAs)* . If an RA is expected by the MN but not received within the interval defined in the RIO, this may indicate a failure of the old AR; therefore, the MN checks the reachability one more time before assuming disassociation from the old AR. If unreachability of the old router is detected the MN sends a *router solicitation RS* and awaits in response a RA from the new AR.

Otherwise, the MN may also receive an unsolicited RA, for instance once it has been offline for a while and then connecting to a new network. In fast-handover proposals in MIPv6, cross-layer techniques allow the IP layer to interpret certain link-layer states directly as movement. A movement detection could be triggered by link change at

layer-2. This, however, can lead to unnecessary layer-3 unreachibility tests if the link change occurred within the same subnet for instance.

Once communication is interrupted for a `DELAY FIRST PROBE TIME` seconds, the node uses *neighbor solicitation (NS)* to confirm reachability with predefined number of retransmissions (`MAX UNICAST SOLICIT`). If no *Neighbor Advertisement (NA)* is received, the MN assumes its has moved. The old router reachability state is set to `NULL` indicating that it is no longer available.

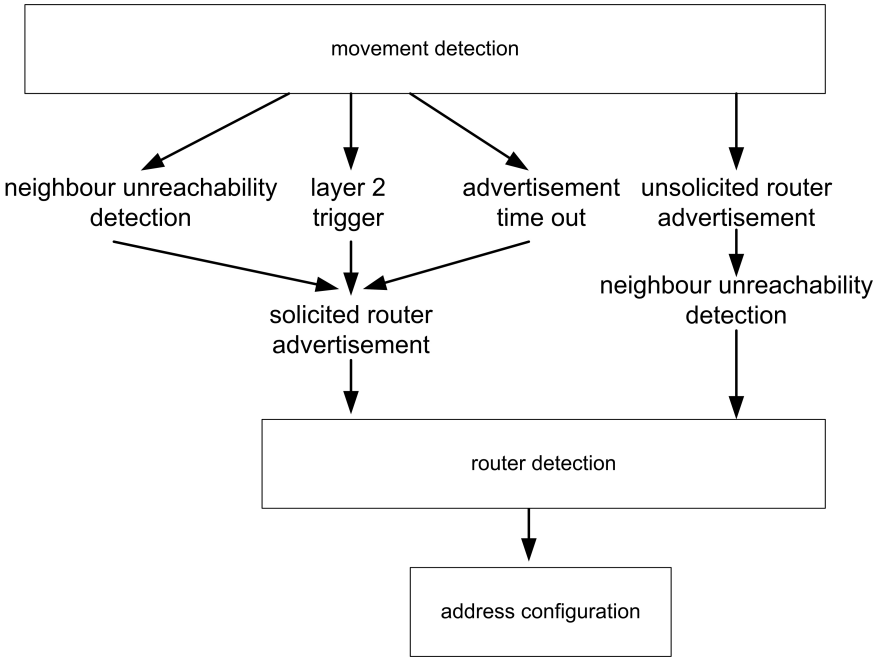

Figure 5.2: Different movement detection algorithms - mobile IPv6

Figure 5.2 summarises the above alternatives of movement detection as given in the original MIPv6 (RFC 3775 [98]).

5.2.1.2 Router discovery

Once connected physically to the new AR, the MN receives either a solicited or an unsolicited RA. This RA includes, among other data, link-layer-related information like the link-layer address of the new router or the MTU (Maximum Transfer Unit) size.

5.2.1.3 Address configuration

The RA sent by the new AR also includes prefix information for auto address configuration and two flags, which indicate whether to use stateless (autoaddressing [136]) or stateful address configuration (using DHCPv6 [63]). The stateless alternative is faster and can be done without explicit approval by the new AR. In the stateless mode, the RA contains a flag inside the prefix information option that is set to autonomous. The MN, then, generates its new *care of address (CoA)* by appending its MAC address or a random cryptographic generated 64-bit number to the router advertisement's prefix.

5.2.1.4 Duplicate address detection

After obtaining or creating the new address, it has to be validated to assure that every node in the network still has a unique address assigned. This is done using neighbour discovery mechanism as described in [135] and is known as *duplicate address detection (DAD)*. Although [98] states that the DAD should be avoided since its associated delay can worsen performance in the case of multiple simultaneous handovers of MNs to the same subnet to prevent desynchronisation. In [117], fast handover techniques are used to allow the DAD procedure to occur before the handover starts. To reduce the latency in the DAD procedure, the new AR constructs a new CoA, then performs a DAD for the MN and stores this new CoA to the *new-CoA (nCoA)* table when anticipating that a handover for an MN is about to happen. This, however, means that not only is the MN anticipating a handover but also the ARs. The authors of [117] also suggest a substantial change in the MIPv6 protocol allowing an AR to exchange MN information with other ARs beforehand. Bagnulo et al. also argue in [119], that DAD procedure is redundant due to the low probability of address collisions.

5.2.1.5 Binding update

After the MN has acquired a valid CoA, it has to inform the *home agent (HA)* and the *correspondent nodes (CNs)* about its new CoA. This is done using a *binding update (BU)* and *binding acknowledgment (BA)* messages. This way, both the binding caches of HA and CNs are updated. To create a BU the *Destination Options* extension of standard IPv6 datagrams is used. A BA is an indication whether updating the binding cache has been successful or an error occurred indicated by specific error codes. Authentication

and public key exchange is made possible protected by IP Security using Encapsulating Security Payload (ESP) [103]. Further investigation of security aspects of MIPv6 and especially the binding procedure are, though, beyond the scope of this work.

5.2.2 Context-Aware Handover in MIPv6

In the last years a lot of work went into improving handover latency in both 802.11 and MIPv6 scenarios (see [44; 49; 52; 102; 116; 117]). A new trend in this area is the use of location information to improve movement detection and proactively initiate handover. The different approaches can be classified in three ways: for what the location information is used, who is implementing the movement tracking and prediction part, and how the location information is used. In the approach followed in this work the location information is used to replace slow link and network layer movement detection processes. There are several other approaches, like [59] and [44], which use location information in order to proactively initiate handover.

A different approach is proposed in [92], where the network is using signal strength together with triangulation to track the MN's location and determining its movement pattern. The network, then, triggers the handover based on the position of the user.

Another network centred approach proposed in [74] predicts future movement using previously recorded movement patterns. Furthermore, the authors of [70] propose using georouting to improve routing between neighbours that are close in terms of location. This way necessary route updates may be handled more efficiently.

The main difference to these approaches is that this work considers location information as part of a multi-dimensional context information, which is managed in a distributed way (introduced in Chapter 3. The network context matching the mobile user context is gathered through a composite overlay to enable an intelligent handover process. The first overlay manages user mobility and tracking of user context. The nodes part of this overlay are called search peers. The second overlay manages and structures the network context into a large scale P2P location-based service. The network context queried and gathered for a given user context is sent to the mobile node to indicate the candidate wireless networks along the user movement path. Network context is also used in the selection process of the most appropriate wireless network and therefore influences the handover algorithm. Furthermore, with location and coverage

information contained in the network context, a predictive triggering of handover can be triggered.

5.3 Managing Network Context in MIPv6/IEEE 802.11 Environment

How context management is carried out for an IEEE 802.11 and MIPv6 environment is explained next. The following sections refer to an implemented framework in the OMNET simulation environment. The goal is to provide a proof of concept of context-aware mobility management in a well known environment. How context is sensed, managed, and exchanged in WLAN and MIPv6 environments is explained.

5.3.1 Accessing a Context Management Framework

There are two aspects that need to be taken into account when enabling context-aware handover: (i) managing the wireless diversity through the P2P framework to allow collaboration between networks, and (ii) allowing mobile terminals to locate the resources and manage the handover.

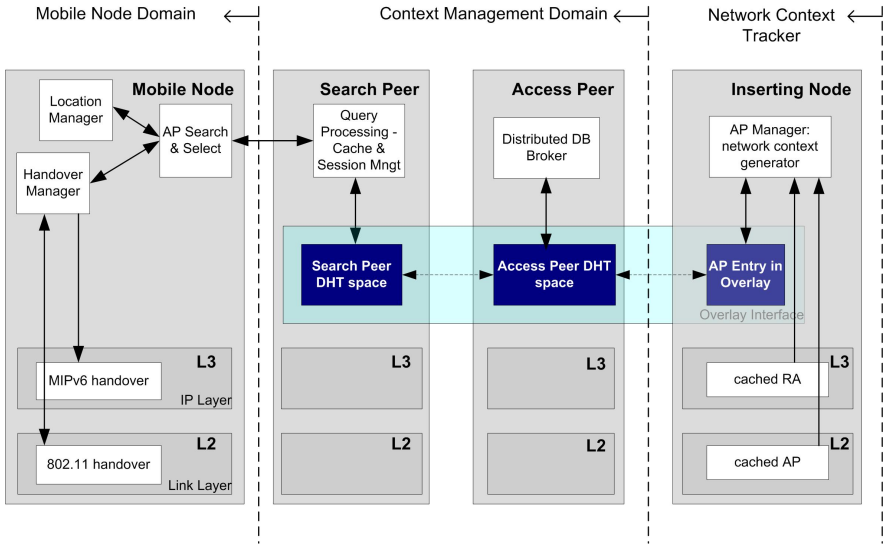

Figure 5.3: Abstract Representation of Overlay Framework - Interaction between operators and users with the overlay middleware

In Figure 5.3, three types of control domains can be identified:

Network Context Trackers refer to how to link network management nodes to the overlay network managing the context. At the operator side, these tracker nodes track the dynamic context of wireless resources (AP online/offline, or saturated, etc), and fill the context file describing the wireless resource. The operator must run an "Inserting Node" on top of a management node which inserts its content into an access peer.

Context Management Domain refers to the network context management overlay introduced in Chapter 3. It consists of "access peer" and "search peers". The access peers must support the DHT interface and are part of a distributed database, which manages network context. Network operators can also run the search peer module on top of dedicated nodes capable of tracking and storing user context such as HLR/VLR.

Mobile Node represents the terminal used by the mobile user during some period of time in addition to user context sensors. The device is assumed to stay the same during the handover analysis part of this chapter. The terminal with its movement tracking or user context-tracking capabilities consist a mobile node. The MN run the intelligent handover part, uses context information such as location to trigger handover (see Figure 5.3). The mobile user registers a mobile node at a search peer. Based on the location tracking capabilities of the user, location updates are sent to the search peer. Query requests are generated by the search peer based on the user movement prior knowledge of movement direction and the dynamics of the motion.

Logically speaking, there are also three separate distributed layers in the architecture illustrated in Figure 5.3, which include:

Database Layer where the logic of the query is processed. At the mobile node *AP search and select* module defines the search parameters according to the movement information and trajectory, and converts this information into a query request sent to the search peer. In the search peer the database element manages a session to allow spacial-temporal processing of query requests sent over the overlay. The replies are filtered out before being sent back to the mobile node. On each access peer, there is a database processing unit to allow SQL type of filtering of

objects that match the request. Also from the network operator point of view, the database layer allows the operator to insert objects (i.e. network context) based on the space model given in Chapter 4.

Overlay Layer which gathers the search peers and the access peers. This layer structures the semantic overlay which efficiently support range queries, spatio-temporal queries, etc. The overlay layer is accessed by network operators to insert data objects stored or managed on the database layer. The user also accesses this layer through a session managed at the search peer.

IP Layer Although the overlay can be used for full message exchange and if an indirection infrastructure is used at the search peers, the whole communication can use i3 overlay to deal with mobility and traffic routing. However, it could still be possible to run mobile IP as normal, so the changes of the CoA are reported to the home agent. Additionally the CoA change has to be reported to the search peer.

The architecture of the distributed system used to track network context and retrieve context information assumes that the user is capable of interpret this intelligence while able to track its own context. At the mobile node and the necessary software elements needed for a prototype context-aware handover management modules are explained next.

5.3.2 Location Tracking at the Mobile Node

Mobility and movement tracking are essential to all mobility protocols independent of the layer or standard used. It can be however said that since layering imposes a separation of concern, the traditional movement tracking is done separately on each layer. However, some mobility tracking mechanisms are more accurate than others. For instance localisation using differentiated signal strength and Kalman filter in UMTS [197], allows an accurate positioning of the mobile node. The same could be said about using a GPS or an assisted GPS to track movement.

The location manager shown in Figure 5.3 deals with discovering the positioning capabilities in the end device. In this implementation, the information is assumed to be provided by a navigation system such as the one provided in modern cell phones or in a

car. The positioning module offers a service to which the location manager subscribes to and therefore defines parameters regarding the frequency to receive updates. This refers to the publish/subscribe mechanisms often used in service oriented architecture.

5.3.2.1 Position Retrieval

A control loop such as a delay locked loop implemented in software is capable of tracking the movement velocity of the user. This means that the update frequency of the messages defined by the sampling period ($\Delta t\ 1/ \propto\ velocity$) has to lock to the velocity change of the user. The delay locked loop updates this within a certain limit and adaptivity which place a low pass filter on short term peak changes. The publish/subsribe relationship between a navigation service and position retrieval is shown in Figure 5.4.

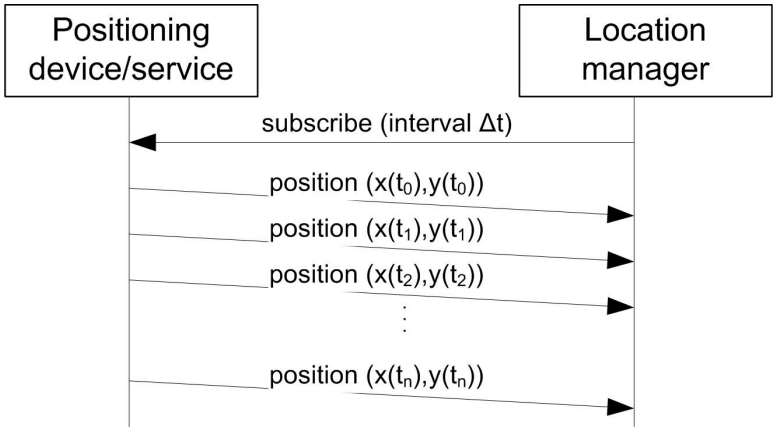

Figure 5.4: MSC Publish rate for location updates - publish/subscribe communication paradigm

The figure shows the position retrieval submodule subscribing to the positioning device. The current position is then sampled at an interval specified by the position retrieval submodule.

5.3.2.2 Overlay Querying

Spatio-temporal queries are defined according to the movement vector of the MN. Given map information such as street network information, the geographic scope of the query can be defined. The goal of the query is to retrieve network context ontology files, which should include configuration information related to the underlying wireless link

ahead of traditional link and network layer discovery mechanisms. The query process depends on the needs of the handover manager, for instance it could be specified similar to a FON map service to lookup any hotspots of a special technology in a radius of a few kilometres. Otherwise, if a path estimation can be planned, the handover manager requires only those access technologies along a movement or planned path. The query process, its dependence on movement, and its overhead have been analysed in Chapter 4.

In the example studied in this chapter, the decision process or selection of the next handover is done with a simplistic location-based algorithm.

5.3.3 Location Aware Handover Algorithm

The location-based handover procedure starts with an active movement detection of the MN using GPS-based positioning module. If movement is detected, a search for the available wireless networks is invoked. The MN then chooses a next network according to position, speed and direction. The studied scenario only concentrates on WLAN AP connected through an IP AR to the Internet. Knowing the coverage position and channel of the AP the MN awaits reaching the coverage area of the AP. The handover manager in Figure 5.3 triggers the link layer handover by forcing the network card to probe just one channel. After association with the new AP at layer 2 is confirmed, and if a new AR has to be chosen, the handover manager, then informs the IP layer to set the reachability state of the old AR to NULL and then passes a cached RA to the IP layer of the MN. This forces the IP layer to immediately switch to the new AR leading to skip the time for movement and router detection.

The selection and triggering algorithm is based on location. The APs are placed along the expected path. A sequence of neighbouring APs is selected using Voronoi graphs [35]. A Voronoi cell of node P is the set of points in space that are closer to P than to any other node [114]. If the Voronoi cells of two nodes share a common border, then a Delaunay edge [114] exists between those two nodes. A location-based handover carried out between two wireless cells which share a Delaunay edge. In Figure 5.5, the trigger initiated at the boundary of the wireless cell right after the crossed Delaunay edge as done in [72]. The decision process could be easily replaced by other multiple criteria algorithms as already explained in Chapter 3 while including other context dimensions

besides location. For the purpose of simulation, only a Voronoi-based handover has been implemented.

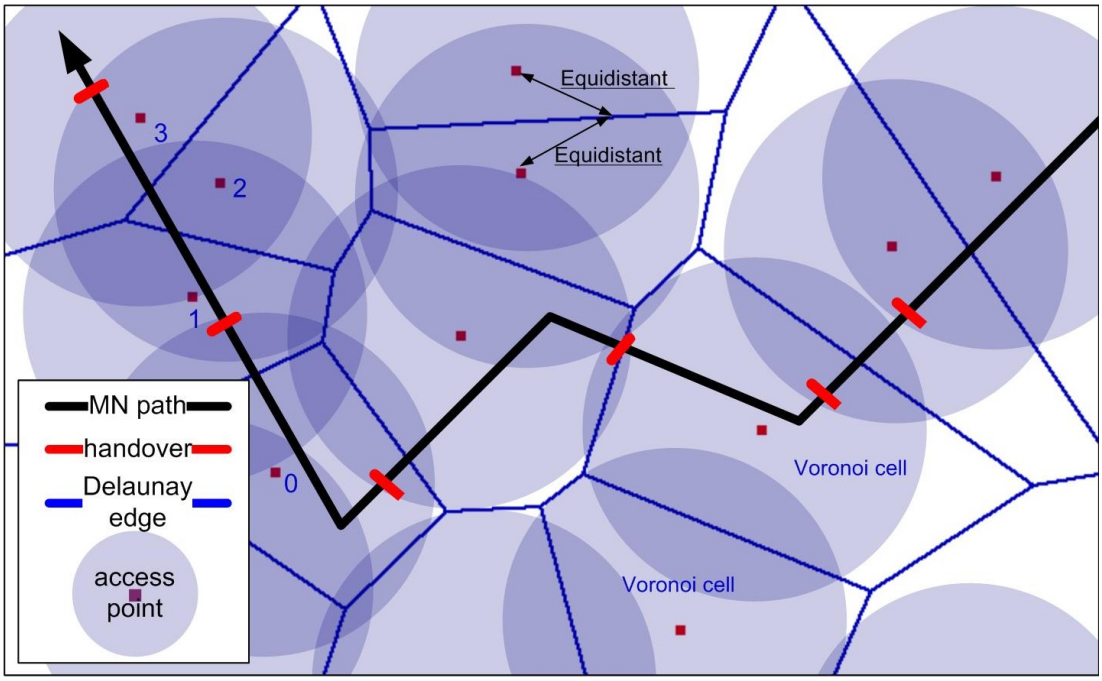

Figure 5.5: Selection algorithm based on the Voronoi graph - purely location-based selection algorithm

The algorithm can be described as follows:

1. Determine current position of MN and APs.
2. Determine neighbouring APs using standard Voronoi graphs.
3. Choose the AP whose coverage area will cover the longest subsection of the MN trajectory starting from the current position, at bootstrapping, or from the ending border of radio propagation range of the current AP and ending at the border of radio propagation of the candidate AP.
4. Determine the crossing point of the radio propagation radius with the trajectory.
5. If the MN has reached this crossing point go back to 2.

It should be noted, that the crossing point may be calculated in different ways. This depends on how precisely the radio coverage is known. If no such information, is

available the radio propagation is assumed to be a circle centred at the AP and having a radius which differs according to the wireless technology used.

The example in Figure 5.5 shows a handover procedure between cells 0, 1, 2, and 3. Once the MN is connected to the AP serving cell 0, the next in line cells 1, 2 and 3 have been identified through coverage information queried via the overlay. Step 2 of the above algorithm can take place producing the Voronoi graph illustrated in Figure 5.5. The next cell is chosen so that it fulfills the algorithm step 3. In fact, this step results in selecting cell 2 rather than cell 1. Given that the boundary of cell 0 crosses the MN trajectory at a point in space, which is located inside both cell 1 and cell 2, the two cells are compared according to the statement in step 3. Cell 2 covers a longer section of the trajectory than cell 1 and therefore it is selected. The next handover needs to be triggered at the boundary of cell 2. The chosen next in line cell is cell 3.

The disadvantages of the Voronoi-based algorithm is that it cannot always cope well with overlapping coverage of heterogeneous cells. An additional selection process is required satisfying other requirements than just location and coverage.

5.3.4 Cross-Layer Interaction with MIPv6

Triggering a handover to a new AP, manually or using software triggers can already be found in operating systems such as $Windows^{TM}$ wireless manager.

Besides the ability to interact with the link layer, a predicted context-aware handover has to involve forced layer 3 handover. The programmed handover manager has to be able to interact with network layer as well. The way to do this is to allow the handover manager to interact with the *neighbour discovery (ND)* state machine introduced in Section2.5.2. This can be carried out when a RA is made available to the ND without waiting for movement detection state followed by the unreachability state (see Section 2.5.2).

The problem with the ND is that it does not expect receiving an RA from the application layer and therefore the latter RA could not be forwarded to ND. Normally, the RA is received from layer 2. An RA is an ICMPv6 message and its role is in discovering layer 3 mobility.

An example of interaction between application layer and ICMPv6 protocol is the Ping6 application. Since a Ping6 utility also injects a request which is transformed

into an ICMPv6 message (echo-request). Similarly a handover trigger uses the ICMPv6 protocol unit to forward the RA massages received from application layer to ND module.

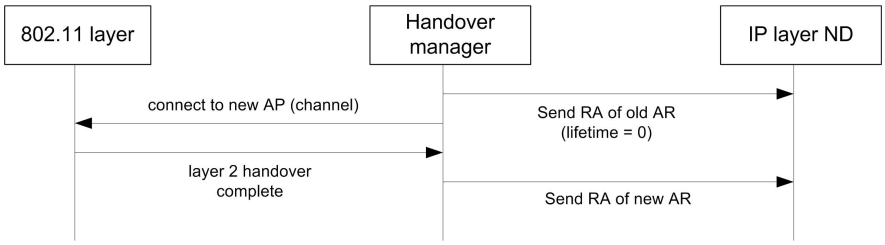

Figure 5.6: Sequence of events to trigger handover at layer 2 then layer 3 - Message type and sequence

Figure 5.6 illustrates the case where a change of both the AP and AR is needed. After connecting to the new AP the handover manager sends a modified RA of the old AR with a the lifetime property set to zero using the modified ICMPv6 link. This forces ND to immediately delete the associated entry of the old AR in the default router list.

The handover manager sends the already cached RA of the new AR to the ND module. The MN has an empty entry for its current default router since the old AR has been already removed from that list. On reception of the RA of the new AR, the reassociation with the new AR can immediately take place without the necessity of NUD with the old AR.

5.4 Implementation with OMNeT++

Today, a variety of discrete event simulation tools for the evaluation of computer networks are available. OMNeT++ (Objective Modular Network Testbed in C++) [34; 184] simulation suite, a discrete event simulation package written in C++, has been chosen to implement the proposed location-based framework. The primary application area of OMNeT++ is the simulation of distributed systems like computer networks. Simulation models are composed of hierarchically nested modules that communicate through message passing. Module functionality is programmed using C++, while the model structure is defined by NED [182], a topology description language. Modules can be combined and reused flexibly, thus allowing the composition of models with any granular hierarchy.

The reason to choose OMNeT++ over other available network simulation tools like NS2 [15] or OPNET [16] has been its clean and accurate modular implementation of the IPv6 suite [17] at the time these experiments have been carried out (i.e. end of 2005). More importantly, the implementation of ND can be considered well developed. At the time of carrying out this work, OPNET 10 and NS2 with their IPv6 suite MobiWAN [21] did not have an accurate implementation of ND nor of ICMPv6.

The goal of the simulation is to implement and test the location-based handover mechanisms introduced earlier in this chapter. Two main setups are chosen to verify the improvement of location-aware handover:

1. Testing performance improvements in comparison with unchanged IEEE 802.11 and MIPv6 handover procedure. The handover procedures are analysed thoroughly under ideal contention conditions and worsened contention.

2. Performance improvement with increasing speed between location-aware handover and normal handover.

The simulation environment has been used to implement the different components that enable location-awareness including the handover manager and location sensing application module. Enabling the application modules to interact with the lower layer in order to trigger a handover has also been the main achievement of this prototype implementation. It can be shown this interaction could be done without changing the underlaying protocol layers.

5.5 Simulation Models

In order to isolate the effect of movement detection, the simulation model is based on a simple topology - shown in Figure 5.7. This topology should optimise the routing of mobile IPv6 packets during handover and minimising the binding phase of MIPv6. Details related to the OMNeT++ implementation of MIPv6 can be found in [113]. The OMNeT++ implementation used in these experiments is compliant to the following IETF RFCs [17]:

- RFC 2373 IP Version 6 Addressing Architecture.
- RFC 2374 An IPv6 Aggregatable Global Unicast Address Format.

- RFC 2460 Internet Protocol, Version 6 (IPv6) Specification.
- RFC 2461 Neighbor Discovery for IP Version 6 (IPv6).
- RFC 2462 IPv6 Stateless Address Autoconfiguration.
- RFC 2463 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.
- RFC 2464 Transmission of IPv6 Packets over Ethernet Networks.
- RFC 2472 IP Version 6 over PPP.
- RFC 2473 Generic Packet Tunnelling in IPv6.
- RFC 3775 Mobility Support in IPv6 (no security).

Apart from that, the suite has models for the following IEEE standards:

- IEEE 802.3
- IEEE 802.11b

To implement an improved handover using MIPv6 with a wireless 802.11 network the following models are of most interest: IEEE 802.11, RFC 3775, RFC 2461, and RFC 2463. The following analysis is based on the following papers [192], [191] and [113] that have been published by the developers of the OMNeT++ IPv6 suite.

The OMNeT++ model implementing the protocol IEEE 802.11 is quite accurate considering the layer 2 part of the protocol. However, the physical radio propagation model and power control are relatively simple [191]. The 802.11 protocol is implemented based on several finite state machines: one for CSMA/CA behaviour, others for whether the nodes take the role of an access point or that of a client. It has been already discussed that the handover procedure in WLAN networks varies according to different AP and client combinations. The developers of the used models made allowance for that introducing two variables: *probe response timeout (PRT)* and *probe energy timeout (PET)*. With these two variables it is possible to model the different behaviour of handover for different vendor implementations, both for the client and the AP. Important aspects of the physical layers like cross talk between channels, have not been modelled

by the developers. This results in a better performance of handover delay measurements, when comparing a similar scenario under simulation with a real setup (as given in [185]).

The model also lacks of the following features: ad-hoc mode, *point coordination function (PCF)*, *wireless equivalent privacy (WEP)*, multi-rate support, power save mode, frame fragmentation, and *request to send/clear to send (RTS/CTS)*. Given that the handover procedure is not influenced by the above techniques, the 802.11 model in OMNeT++, is considered to be sufficient.

On the other hand, the modelled ICMPv6, ND, and MIPv6 have been implemented very accurately. The developers of the IPv6 model have nested the ND model inside the ICMPv6 model. An explanation for this can be seen in the way the ND RFC defines new ICMPv6 messages. Therefore, any ICMPv6 message that is not part of the core ICMPv6 message definition will be forwarded to the ND module for further processing. In the ND module, a distinction is made between routers and normal hosts. According to the ND RFC the ICMPv6 messages have to be processed according to the state of the node. To achieve this, an object orientated class inheritance approach is used, by the developers of IPv6 OMNeT++ package [113], in order to implement the different behaviour of routers and hosts. This technique is also used to extend ND with the specific requirements of MIPv6. To implement the different roles that MIPv6 introduced, a special mobility class and derived classes have been implemented to represent whether the node is a MN, CN, or HA.

According to the developer of these models the following features are not implemented: return routability procedure, full proxy neighbour discovery, improved robustness of handover process when the mobile node returns home, dynamic home agent address discovery, and router renumbering at home.

5.5.1 Modeling Location-Aware Handover

The location-based handover framework has been implemented as OMNeT++ modules according to the interaction explained in Section 5.3 (see Figure . The overlay part is not implemented, and the network context is assumed to be available at the mobile node at the start of each simulation run. This consists in a cached RA and cached AP properties related to AR2 and AP2 respectively (from Figure 5.7).

The *Handover Manager* module (shown in Figure 5.3) receives the current position of the MN through the *Location Manager* module on a continual basis. If the current position of the MN is equal to the *locationHandoverX* parameter and the *location_ aware* parameter is set to the *Handover Manager* triggers the location-based handover. Otherwise, the handover procedure takes place according to the WLAN and MIPv6 protocols.

The location-base trigger interacts with two layers. At layer 3, the ICMPv6 module in the MN receives a message from the handover manager module indicating to clear the old AR entry in its ND module. Then, a *start handover message* is sent to the link layer software module indicating the channel and forcing a channel scan. After the MN receives a signal that indicates that 802.11 handover has been successful, the *Handover module* sends the cached RA of the new AR to the ICMPv6 module.

It is important to note that no changes to the actual ND module and the way how probing in 802.11 is simulated were made.

5.5.2 Chosen Scenario Description

In Figure 5.7, three 100 MBits/sec Ethernet-based subnets are connected using three routers. One access router is assumed to be the home agent (AR1), while the second (AR2) is a visited one. The two subnets include, each, an 11 MBits/sec IEEE 802.11b access point. The MN, initially attached to AP1, is assigned an IP address (by AR1) using Automatic IP [119]; the server on the other hand, has a static address assigned. Each router broadcasts to its neighbours a RA with an inter-arrival delay exponentially distributed with a mean of 1.25sec. The server runs two applications: a Ping6 and a UDP stream server. The UDP server is streaming one 600bytes datagram with inter-arrival delay exponentially distributed with a mean of 0.075sec. This results in an average data rate of 640kBits/sec, which is a comparable rate to receiving a live video on a PDA screen for instance.

Figure 5.7 illustrates the simplistic implementation location-based handover in the OMNeT++ environment. The experiment is conducted with a MN travelling at a speed of $3m/sec$ along a distance of 190meters. The MN is assumed to be already associated with AP1 and AR1. The travelled distance is chosen in a way to cover the radius of the radio coverage of both AP1 and AP2. Both APs have a radio propagation radius of about $85m$.

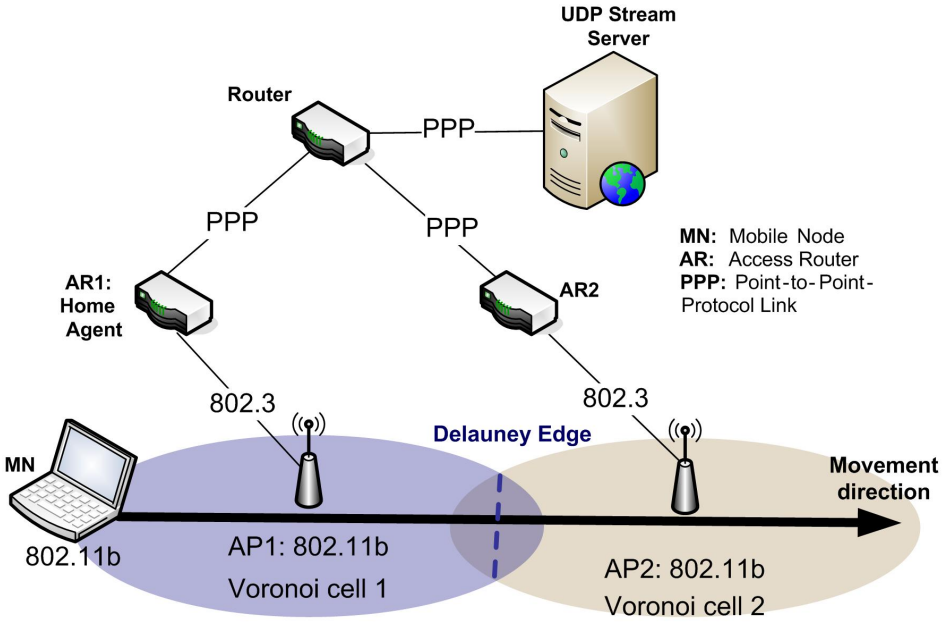

Figure 5.7: Simulated topology layout - context aware handover implementation in OMNeT++

Every setup is simulated several time with a different random seeds to gather statistically significant data. Although the velocity of movement is kept constant at each setup, the randomness of RA frequency is shown to affect greatly the handover delay. The discrete-event simulation uses stochastic sojourn time in each state. The unchanged mobility detection protocols at both link and network layers include a sequence of interrelated discrete states, such as the case in ND state machine (explained in Chapter 2). The modelled network components (routers, access points, etc.) have been set to include infinite buffers. This allows to avoid any losses due to buffer overflow. The only loss of packets that occur is due to badly addressed packets (i.e. packets addressed to a CoA which is no longer valid). Movement of a node is deterministic while most events in each node are stochastically generated. Other aspects such as collision can be excluded when using different random seeds. A higher confidence interval can only be achieved with several runs. This is discussed later in more detail (see Figure 5.9).

The first scenario compares the handover delay for setups where location-based triggers are incrementally enabled. The same setups are then tested in the presence of contention. The second scenario analyses the performance advantage when using higher velocities.

In the location-based case, the MN triggers handover based on its movement and position knowledge. The Voronoi graph with the two cells is rather simplistic. The first scenario consists in measuring handover events for the same travelled path with the same constant velocity.

The duration of a simulation run is taken as the time on which the node reaches a given position on the path. This duration is kept constant for the first scenario, but varied according to the simulated speed of the MN in the second scenario. The instance at which each event of interest (defined in Table 5.1) occurred in simulation time is measured in each simulation run. I denote $T(E_n)$ as the time recorded when event E_n occurs. L2 refers to the link layer related events and L3 refers to network layer related events.

Event of interest	**Event type and number**	**Event definition**
Threshold too low	L2 Event E_1	**802.11**: time of loss of connection to old AP
Restart active scan	L2 Event E_2	**802.11**: time of restarting scan
Probe channel	L2 Event E_3	**802.11**: time to probe one channel
Start authentication	L2 Event E_4	**802.11**: start of Authentication, Authorization and Accounting (AAA)
Layer2 Trigger	L2 Event E_5	**802.11**: end of 802.11 handover
New RA	L3 Event E_6	**MIPv6**: time to receive new RA from new AR
Movement detected	L3 Event E_7	**MIPv6**: time of movement detection
Start BU	L3 Event E_8	**MIPv6**: beginning of binding update
Handover complete	L3 Event E_9	**MIPv6**: first ping after handover completion

Table 5.1: Overview of the measuring points - measures the event's occurrence in each simulation run (in seconds)

5.6 Simulation Results

5.6.1 Delay Analysis of Context-Aware Handover vs. Existing Non-Improved Handover

Figure 5.8 records the sequence of the measured event instants for the location-based handover and non-improved handover as defined in Table 5.1. The graph in Figure 5.8 shows a snapshot of both types of handovers. For type of handovers the event instants are plotted vs. time. Two parallel lines result (i.e., location-aided, shown as *"improved"* on the upper timeline, and standard WLAN handover followed by MIPv6 handover according [183], shown as *"standard"* on the lower timeline).

In Figure 5.8, the handover latency is brought down from 1850msec to only 125msec using the location triggers for the combined two layers. In the standard case, probing all WLAN channels, followed by detecting a new router advertisement on the new channel engulfs a considerable portion of the overall delay, as expected. Another important result is the considerably reduced number of handshakes and probing events, in the improved handover case.

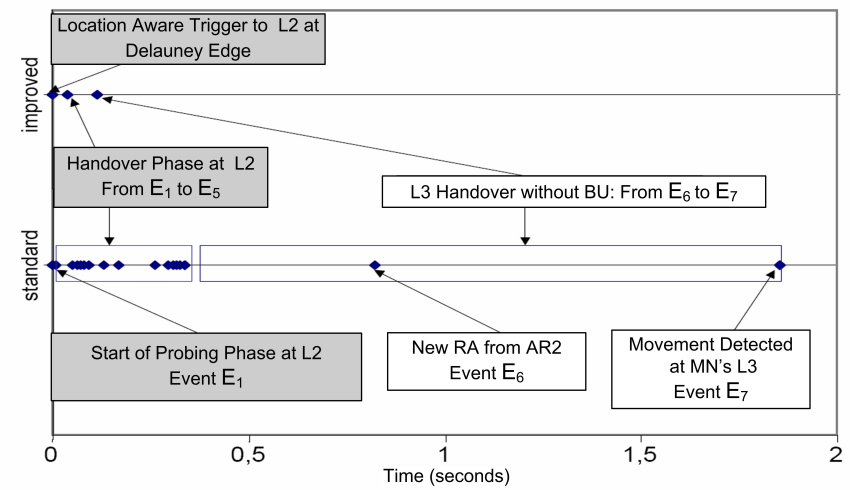

Figure 5.8: Timeline of a single simulation run recording event occurrence *"standard"* vs. *"improved"* (location-aware handover) - L2 stands for handover events at the link layer and **L3** handover refers to the measured MIPv6 events (listed in Table 5.1)

Taking the results of the 200 simulation runs, distributions of the major handover phases at both link and network layers relating to movement detection and handover are

given in Figure 5.9. The figure shows the delay distribution of each of the major phases of the link layer handover and the MIPv6 handover when disabling the location-based triggers (i.e., the standard handover).

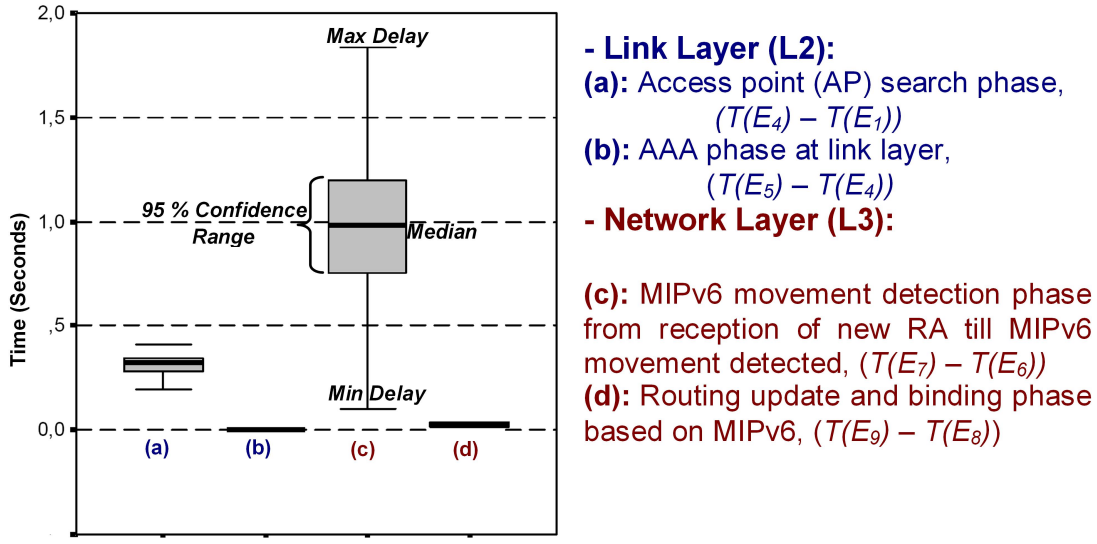

Figure 5.9: Delay distributions of major handover phases with no location-aware triggers - OMNeT++ handover implementation without any changes

The delay distribution of the main phases for both link layer and network layer movement detection and handover process are discussed. In Figure 5.9, the search phase at the link layer ($phase(a) = T(E_4) - T(E_1)$, defined in Table 5.1, produces the largest part of delay as part of the link layer handover (with an average of 350msec). The Authentication, Authorization and Accounting (AAA) delay ($phase(b) = T(E_5) - T(E_4)$) is at least a degree of magnitude lower than the real values measured in [52], due to the rudimentary AAA implementation under OMNeT++.

Now taking a look at the MIPv6 delays, the delay incurred before movement is detected is measured as ($phase(c) = T(E_7) - T(E_6)$). Movement detection is the most significant part of the delay, under the assumed simulation setting. The phase is measured from the instant when an RA is received by the MN from the new AR (which can only occur on the new WLAN link to AP2) until MN's ND module changes the state of the currently registered AR from the old AR (AR1) to the new AR (AR2). The binding update then follows ($phase(d) = T(E_9) - T(E_8)$) takes a much shorter time, remembering the simple topology chosen for this purpose. The movement detection at

the IP layer (phase (c)) has a large variance due to the 1.25sec RA interval. If the MN receives an RA on the newly established link to the new router, right after completing the handover at the link layer, the IPv6 movement detect phase could complete right away in some cases (minimum delay in phase (c)). This effect is also analysed in Figure 5.10.

The latter figure shows the impact of location-based triggers on the WLAN handover and that on the MIPv6 handover separately. It compares four scenarios, where each of the link (L2) and network layer (L3) triggers are turned ***"on"*** and ***"off"*** interchangeably. The cumulative distribution function (CDF) of the final handover latency, applied to the collected results of each of the 200 runs show the significant improvement of Location-aware triggers in shortening the delay and decrease the large distribution of the final delay (see Figure 5.10).

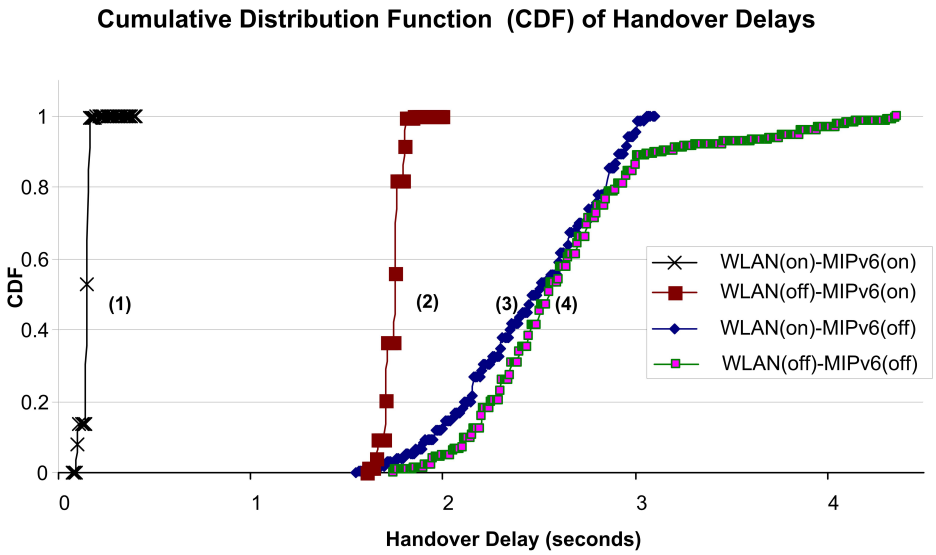

Figure 5.10: CDF of final handover latencies from loss of first ping response to resumed communication - ***"on"*** refers to the enabled location-aware triggers at the Link Layer (WLAN) or Network Layer (MIPv6); - ***"off"*** refers to disabling the triggers; Scenarios: (1) WLAN "on" - MIPv6 "on"; (2) WLAN "off" - MIPv6 "on"; (3) WLAN "on" - MIPv6 "off"; (4) WLAN "off" - MIPv6 "off"

The final delay distribution plotted in Figure 5.10 is taken to be the lapse of time between the first loss of communication and the instance at which communication resumes, i.e. $T(E_9) - T(E_1)$. For this purpose frequent Ping6 messages are sent every

10 msec from the MN to the *corespondent node (CN)* (server) to simulate a highly interactive application. No major propagation delay is assumed. Analysing the obtained delay distributions, it can be seen that only the combination of location-based triggers being used at both layers can improve the handover significantly (Scenario (1)).

The worst latencies are measured under Scenario 4, where the total handover delay is the largest. The delay also differs greatly from one trial to another, which is due to the large beaconing interval of the standard MIPv6 RA interval. As a matter of fact fixing the rate of beacons sent at either the link layer or the network layer is a matter of cost vs. accuracy in detecting the movement. The assumption in this OMNET++ simulation model is that the lack of any cross layer communication, forcing each layer to detect movement on its own. The routers send RAs every 1250 msec (exponentially distributed [72]). This process follows a stochastic distribution. The effect of the RA interval shows up in both scenarios (3) and (4) ("MIPv6 off"). This effect is nonetheless totally removed once looking at scenarios (1) and (2), where the "MIPv6 on" trigger is used.

5.6.2 Contention Effect at Link Layer

The effect of contention leads to a more realistic behaviour to the way signalling frames access the shared wireless medium. The expected result is a slight increase in handover delay. Therefore, it is important to test the performance of the algorithm in the presence of contention.

5.6.2.1 Setup

To simulate extra load, two non-moving wireless nodes are configured to communicate using the visited access point AP2. The two additional nodes receive each a downstream CBR UDP flow originating from the same UDP server accessed by the MN. The MN is set up to use location-based handover.

5.6.2.2 Results

Two scenarios are compared, the single-node scenario against the 3-nodes scenario. The two scenarios are run 200 times with different seeds. With CSMA/CA a 'listen before

you speak' approach is used resulting in a time slotted medium access. Since the back-off process in CSMA/CA is based on a stochastic process, the different runs reflect well the random nature of the total delay. The total handover delay for both scenarios is compared in Figure 5.11 using cumulative distribution function.

For the single-node scenario, it can be said that two groups of results arise. In around 55% of the measured 200 cases, the handover delay lies below 11 msec, whereas in the remaining group of cases, the delay lies around 36 msec.

The second scenario results, on the other hand, show that all of the delay measurements are above 36 msec. This can be explained that there are two main slots involved in the two scenarios. Since the second scenario is more realistic (where several nodes exist), the effect of contention can be said to result in about 25 msec additional delay. In fact, since the CSMA/CA transmission slot assignment is a stochastic process, even for the single-node scenario, 45% of the cases use the second time slot to access the channel rather than the first one.

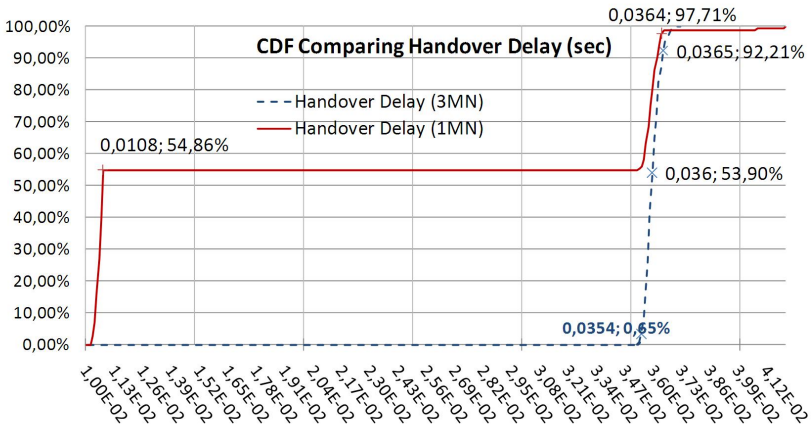

Figure 5.11: Location aided handover delay with different numbers of attached nodes - CDF illustration vs. delay in seconds for the two scenarios

One more difference between the two cases can be said about the start of each of the slots. In the 3-nodes scenario, since the shortest ever delay of 11msec is never achieved, this means that in all measured cases, a backoff process has to occur. The simulated time that elapses before access is granted varies from each simulation run to the next. Therefore, when zooming in the distribution of the handover delay in the 3-nodes scenario (see Figure 5.12), a Gaussian distribution can be noticed. In other words,

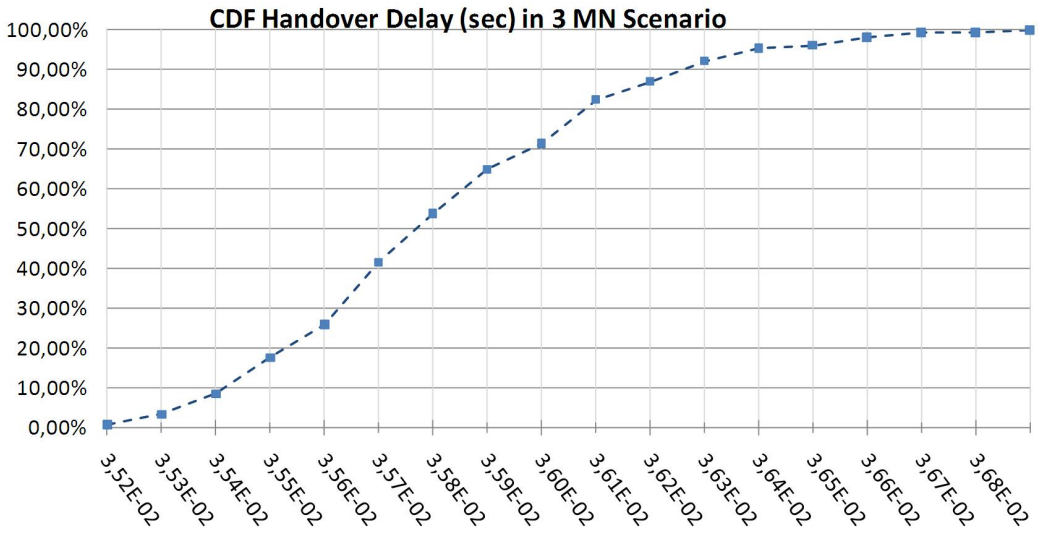

Figure 5.12: CDF of handover delay in the contention scenario - Distribution of the 200 cases of measurement points looked at in detail

the results reflect the total handover delay measured with 99% confidence interval to be (35.8 ± 0.0677) msec.

5.6.3 Impact of Movement Velocity

5.6.3.1 Setup

The idea behind the last experiment is to test the performance of handover with a MN travelling at different speeds. The setup is similar to the first scenario, where a single node travels between two Voronoi cells, with the difference that only two types of handover are compared: standard MIPv6 handover with a frequency of 1.25 RA per second and location-based handover with the functionality of improved channel search and the use of cached RA both switched on. These two handover scenarios have been conducted with a MN traveling at three different speeds: 3 m/s ($\approx$ 10 km/h), 15 m/s ($\approx$ 50 km/h) and 30 m/s ($\approx$ 100 km/h).

These six different scenarios are again simulated 200 times with different random seeds. The simulated time periods have been adapted to let the scenarios run for the time needed for the MN to cross the same distance (see Table 5.2 for details). At the beginning of every run, the MN is set to wait 5 seconds in order to assure that the MN

has successfully connected to the first AP; the measured data within these 5 seconds are later subtracted. In fact, in all scenarios, the same rate CBR UDP stream is used, but the total number of transmitted datagrams depends then on the simulated time period.

Speed	Simulated time	Path length
3 m/s	60 s	180 m
15 m/s	12 s	180 m
30 m/s	6 s	180 m

Table 5.2: Speed versus simulated time

5.6.3.2 Results

To be able to compare the three different speeds, a ratio of the number of UDP datagrams that are received during each of the scenarios divided through the total number of datagrams destined to the user (i.e., including all datagrams that are generated at the server after the first static 5 sec) is measured. This represents a success ratio of each setup, which is calculated using the median of the measured data as follows:

$$Success\ ratio = \frac{Number\ of\ Received\ Datagrams}{Total\ Number\ of\ Sent\ UDP\ Datagrams}. \tag{5.2}$$

A first glance at the data (see Figure 5.13 for details) reveals an expected relation between speed and received data for a MN traveling the same distance at different speeds. The faster the MN, the less time it has to receive data and therefore the more important the handover delay gets. In addition to that, a higher handover delay has a greater impact in a scenario where the MN moves at a higher velocity.

The improved handover manages to deliver at the higher speeds (30 m/s) almost double the number of datagrams the traditional handover achieves. The worsening trend would then continue for higher speeds, before reaching a limit where WLAN access should not even be selected by a context-aware handover due to the short handover distance it offers when compared with the travelling velocity.

5.7 Chapter Summary

In this chapter a simplified version of context-aware handover has been implemented. This chapter focused on the functional integration of a context-aware handover mech-

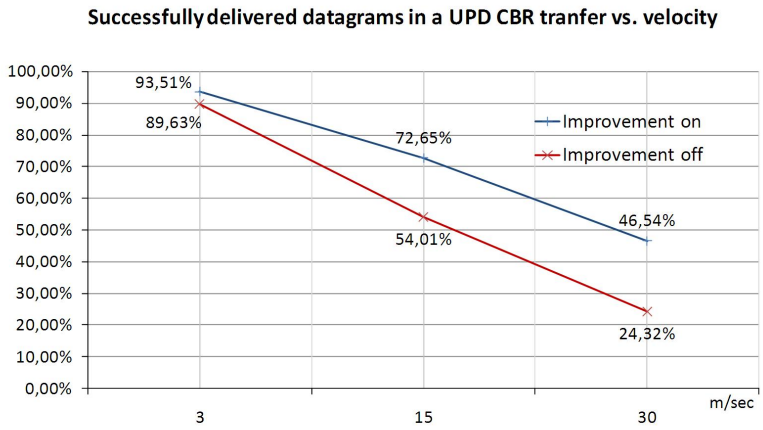

Figure 5.13: Comparison of speed with success ratio representing the proportion of successfully received UDP packets over that of total sent packets - improvement "on" represents the location-based handover scenario, whereas **improvement "off"** represents the standard WLAN-MIPv6 handover scenario

anism in an intelligent mobile node. The MN in fact is able to track its position and location quite accurately and therefore could base its context-aware handover on location-awareness.

The example studied in the chapter only focus on location-awareness as one important context dimension. So the handover management is focused on selecting the most appropriate wireless cell along the movement path. The chosen selection algorithm is based on a Voronoi graph, which purely selects cells based on their overlap with the predicted movement path.

To analyse this simplified context-aware handover algorithm (which is reduced to a location-based handover), a network environment is also selected for this purpose. The chosen technology is MIPv6 and WLAN as wireless domains. For these wireless technologies a context-model is extracted. It is shown in the chapter that it is sufficient to describe WLAN-IPv6 cells with three types of context data: *(i)* exact location of radio coverage, *(ii)* wireless channel used by the wireless AP, and *(iii)* cached router advertisement describing IP-layer information related to the serving access router of each wireless domain.

The network context description is extendible, but since a focus is made on implementing a simulative prototype of the framework, it could be said that it is sufficient

to use mentioned static context information. Other context, such as load, trust, bandwidth, etc, has been excluded from this case study.

Another major contribution of this chapter is the feasibility of a context-aware handover triggers, which interact with lower layers in the mobile node, after an intelligent selection process has completed. The interaction with the WLAN and MIPv6 protocol entities has been demonstrated in this case study. For this a thorough understanding of the functioning of each protocol has been needed. For a location-based handover, the mobile node can then trigger a handover on demand.

It turned out that only a few changes have had to be made to the underlying link and IP layers models. The context-management system has been set to sense information about the access networks. The MN retrieves network context, which include a cached RA and the wireless channel's characteristics (such as modulation and frequency band). The MN forces its link layer (through operating system calls) to scan a given channel for a new AP, and then triggers its own ND through two messages RA with a lifetime of zero to skip the movement detection from the old AR; and a cached RA from the new AR to initiate the location-based handover at the IP layer. The resulting programmable handover is much simpler than the one proposed by the IEEE 802.21 media independence. However, it is limited by the ability of application modules to access network drivers and communication protocol entities. For IPv6, the ICMPv6 protocol entity is rather easy to access, but the link layer might depend on the operating system.

Performance comparison of the WLAN-MIPv6 standard handover procedure with the location-based handover, has shown the considerable improvement in terms of handover delay. The exact performance gain of enacting location-based triggers for either WLAN or MIPv6 layers has been shown. The combined triggers perform best especially for higher velocity scenarios.

Finally, it is worth noting that the skipped probing and beaconing at the two layers saves unnecessary signaling during a critical phase of lower connectivity. In the context-aware framework, beaconing, closely before to the handover phase has to take place, in order to discover the wireless diversity is considerably reduced. This is, however, replaced by an overlay signalling which needs to take place ahead of the handover. The network context that describes nearby cells is transferred to the MN before the wireless channel degrades. The handover is then triggered in a preventive manner rather than

forced by a lengthy channel degradation. Also beaconing during that phase is avoided since it might carry a large energy footprint for a MN.

The effect of worsening bandwidth and channel quality when reaching the edge of a wireless cell, has not been included in this study. This requires a much more accurate radio propagation model, and an accurate RTC/CTS OMNeT++ implementation of bandwidth reduction [193].

Chapter 6

Evaluation and Discussions

This chapter aims at evaluating the architectural aspects of integrating context-aware mobility management in the future wireless Internet. The choices made throughout the thesis are revisited to define the related issues and open questions that need further investigation.

6.1 Context-Aware Mobility Management Architecture

The context management chain consisting of sensing-reasoning-acting on context information is seen as being part of the mobility management of the future integrated wireless world. To start with, mobility management in 4G network has been broken down in key and basic functions that entail dealing matching user behaviour and application needs to the surrounding wireless diversity supposed by 4G systems. Context-awareness offers the way to implement these basic mobility management functionalities independent from the different underlying technology platforms, while focusing on the user's situation in order to enable the *always best connected* vision, rather than just continued connectivity, which is the case in today's mobility solutions.

The context management chain has been shaped to deal with mobility. First, sensing network heterogeneity, on the one hand, and the user's situation and activity, on the other, can be said to replace parts of the location tracking, and beacon and signal measurements found in a traditional mobility management architecture. The context reasoning part is done in the form of a decision making process that is meant to maintain the "always best connectivity" based on, sometimes, partial or fuzzy information. The

decision process has to provide a way to reach a compromise between several conflicting objectives through the use of multiple criteria selection. The resulting decision is then executed through the triggering of vertical handover procedures, in a way that caters for heterogeneity of lower layers.

The architectural choice to integrate the context management chain has been carried out mostly on the application layer throughout the thesis. The network context sensing for each mobile user has proved to be a challenging endeavour, which has to be scale. As a result, and in order to support context sensing and collection in a mobile environment, Chapters 3 and 4 develop a distributed mobile location-based service, used to manage network context. The network sensing and collection process is adaptive to the user context.

The self-awareness and context tracking capabilities of the user can be very different depending on the hardware and software functionalities found at the end device. How user context can influence the validity and quality of the context management chain has been shortly discussed in Chapters 4 and 5. For instance, the movement detection technology available for the user could greatly influence, which part of the collected context information is taken into account during the reasoning phase. Also the triggering of anticipated handovers can be limited by the localisation technique used at the mobile device. Some examples of how localisation techniques can influence the decision process are discussed next.

1. Accurate localisation, enabled through built-in dedicated hardware like a GPS receiver [95] or assisted-GPS [65] to allow indoor accurate detection. The selection process can occur in advance for the predicted paths.

2. Less accurate localisation: includes the situation where no GPS signal or other indoor localisation technology is available. The possibility to use Place Lab [10; 85] for instance. The advantage of the Place Lab is that it is destined to work on top of any wireless technology where a fixed infrastructure can be beaconed by the mobile user. The place lab works both outdoors and indoors without relying on a single type of signal (as is the case for GPS) nor does it require dedicated physical layer process such as in UMTS localisation or assisted GPS. The beacons received from any base station (e.g., GSM, or UMTS) or access point (e.g., WiMax or WLAN) is checked against a database that stores to each access point or base

station geographic coordinates, based on a localisation software, the place lab can locate the user with an accuracy of about $10m$ to $20m$.

3. In the case of the absence of an accurate localisation technology, similar to the place lab, it is still possible to approximately deduce the location of the user with the following rule "first discover the wide area cell geographic coordinates then define a search diameter limited by the search diameter of the single wide area cell". In the case of receiving several beacons, the place lab is shown to achieve an accuracy that is capable to support some pre-configuration efforts or negotiation with future wireless hops.

6.1.1 User Context Representation and Service Oriented Architecture

The assumption of the context-aware framework is to be able to take advantage of available context. On the user side, context is meant to collect and describe the user's capability to be self-aware, i.e., tracking his/her own situation. The modularity and generic approach can be achieved by describing different user capabilities through a *service oriented architecture (SOA)*, for instance. *Universal plug and play (UPnP)* is example of how dedicated small devices can be described according to the services (as in software services) which they might offer for other applications. Another alternative is using semantic web technologies (for instance OWL-based) to describe incrementally extending user context to include device capabilities and their role in the architecture. It is important to note that this thesis does not really focus on the description language or software platform that could most suit the semantic data modelling, but has to offer ways of how this capabilities can be used.

The example of localisation techniques should be semantically describable in way, that for instance the GPS receiver and route planner capability can be detected leading to a tracking and decision mechanism similar to that explained in Chapter 5. But if the GPS signal becomes available, then the localisation technique might change relying on Place Lab software. In this case, the role of location prediction becomes less important in the decision criteria, and a more reactive connectivity maintenance should be followed, while taking advantage of available more reliable context information.

6.1.2 Cost of Integrated Location Tracking, Example UMTS

In existing cellular systems, the distance between the mobile and a given base station (BS) is observable. Such information is called the forward link received signal strength indication (RSSI) signal, which is available in both global system for mobile communication (GSM) systems and code-division multiple-access (CDMA) systems for transmission control. The RSSI signal is the average of the beacon (or pilot) signal strength received at the MNs. Although the received signal strength is affected by propagation attenuation, multipath fading, and shadowing, the RSSI signal is only related to propagation attenuation and shadowing since the rapid fluctuation of multipath fading is removed by the averaging operation. Denote $p_{k,i}$as the RSSI signal received by a given MN from the i^{th} base station (BS) at time t_k. This received signal power (measured in decibels) can be modelled as a function of the distance plus the logarithm of the shadowing component [118], i.e.,

$$p_{k,i} = p_{0,i} - 10\eta \log_{10}[(x_k - a_i)2 + (y_k - b_i)2]^{1/2} + \upsilon_{k,i} \tag{6.1}$$

where $p_{0,i}$ is a constant determined by the transmitted power, the wavelength, and the antenna gain of the $i^{th}BS$; (a_i,b_i) is the position of the i^{th} BS; η is a path-loss index (typically,η=2 for rural environment η=4 and for urban environment); and $\upsilon_{k,i}$ is the logarithm of the shadowing component, which is modelled by a zero-mean Gaussian random variable with standard deviation of 4 - 8 dB [154]. The equation 6.1 captures the RSSI measurement at time instance i. At moment $i+1$ the user would have moved according to a continuous dynamic motion equation given by the state of the MN at time t is given as the column vector [197]:

$$s(t) = [x(t),\ \dot{x}(t),\ \ddot{x}(t),\ y(t),\ \dot{y}(t),\ \ddot{y}(t)]^T, \tag{6.2}$$

where $x(t)$ and $y(t)$ specify the position, $\dot{x}(t)$ and $\dot{y}(t)$ specify the velocity, and $\ddot{x}(t)$ and $\ddot{y}(t)$ specify the acceleration in the x and y directions in a two-dimensional grid. The state vector can be written as

$$s(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{bmatrix} \tag{6.3}$$

where $\mathbf{x}(t) = [x(t),\ \dot{x}(t),\ \ddot{x}(t)]^T$ and $\mathbf{y}(t) = [y(t),\ \dot{y}(t),\ \ddot{y}(t)]^T$. The acceleration vector, $\mathbf{a}(t) = [\ddot{x}(t),\ \ddot{y}(t)]^T$ is modelled as follows:

$$\mathbf{a}(t) = \mathbf{u}(t) + \mathbf{r}(t) \tag{6.4}$$

$\mathbf{u}(t) = [u_x(t), u_y(t)]^T$ represents a discrete command process representing sudden changes in acceleration levels, whereas the $\mathbf{r}(t) = [r_x(t), r_y(t)]^T$ is a zero-mean Gaussian process chosen to cover the gaps between adjacent levels of the process $u(t)$. What is important to note is that the model gathers changes in direction through the x and y components of the vector. In order to create realistic user behaviour [118] and [197] go into more details about the stochastic models used to simulate user movement change. They also both use some filter design to track the user movement based on RSSI measurements. In [118] the tracking of user exact movement is used to predict handover across ATM cells, and therefore allows provisioning of resources in advance. In [197] a Kalman filtering is used to track the MN based on the RSSI measurements with the goal to predict the trajectory of the user in advance.

By sampling the state vector every T time units, the motion vector can be described in terms of the discrete-time state vector $s_n = s(nT)$. The frequency at each signal has to be measured should be high enough to accurately track location in fast movement cases. Localisation is generally less costly in terms in energy usage than discovering wireless connectivity or spectral diversity. This is due to the fact that energy can be measured passively at the mobile device level rather than beaconed actively. Therefore, the approach followed in this thesis suggest separating the location tracking effort from the discovery of the wireless diversity.

6.2 Discovering Wireless Heterogeneity

The advantage of shifting the discovery mechanism to an offline process, based on a pre-discovery of network context, and on predicting near future, the beaconing is carried out on the backbone level of the network and not between the network edges and the MN.

This means that discovery and processing of the context, can be done in a semantic way as a background application that uses any of the active wireless links to discover and communicate network context. This is done with the assumption that some prediction

of the future positions can be done to allow the context-aware framework to process the data in advance.

The associated discovery cost is placed in the self-organising overlay network, which manages network context and user search sessions. Any responses are sent to the user as network context that matches their own context. An additional cost is in the computational footprint needed at the user side. This is due to the following additional functionalities, needed to deal with handover management in heterogeneous wireless networks:

- Selection algorithms are needed for both filtering useful network context and identifying next in line wireless cells, which satisfy multitude of both performance criteria and user preferences (called user context). Since the retrieved network context might be fuzzy, a fuzzy-logic selection algorithm is more appropriate. However, the associated computational overhead could be too high for a handheld device. Therefore, selection algorithms, although require context information to reason in real-time, the simpler the selected algorithm the closer it is to realising continuous selection in real-time (e.g., the Voronoi-based algorithm suggested in

- Location-tracking capabilities need to be discovered in each device. As a result the location-tracking is meant to deliver real-time knowledge about the user location at the application level. This information is used for two main processes: (i) user context definition at the query definition stage, (ii) selection of next-in-line wireless networks and triggered timely handovers. According to the location-tracking capabilities of the end terminal, a map with street networks is assumed to also exist in order to correlate current user position with future predicted locations.

- Triggering a context-aware handover results out of the selection algorithm and, therefore, based on the context-aware decision made. The lower layers at the MN have to be triggered to execute the handover from the application layer. Adding an interface to address these lower layers is not trivial and requires a further standardisation effort similar to the IEEE 802.21 standard.

6.2.1 Energy Footprint

The assumption about discovering the network context is that it is not possible to track signal strength in heterogeneous environments, without dedicating several interfaces to beaconing or signal strength measurements. A radio-based handover triggering mechanism relies mostly on a continuous signal measurement that leads to a handover decision (at least as this is the case for call continuation in the cellular network architecture). Even if listening to beacons of nearby access points and base stations is appropriate when using the Place Lab software, the latter only requires a list of nearby access points and their identity to correlate with a separate geographic database. This can also be supported by the offline context discovery using overlays.

With the querying of network context, the wireless diversity is discovered without relying on continuous spectrum scanning, which is a very costly process. Instead, the separation of location and movement detection from the discovering wireless diversity is more energy efficient. One the one hand, enhanced localisation techniques, can be employed as part of sensing the user context. While on the other hand, the discovery of the wireless diversity is carried out mostly in an offline query process.

The benefit of this design principle is that it can be optimised according to the user and application needs, while reducing the conversational negotiation between wireless interfaces and the surrounding wireless diversity, which often is more costly in term of energy than the additional computational overhead suggested in this thesis [127; 148; 156].

Wireless spectrum scanning is also addressed within other research efforts working on developing smarter physical layer techniques especially related to agile and software-defined radios.

6.2.2 Software Defined Radios

The term *software defined radio (SDR)* [18], as the name suggests, defines those efforts to create a software description of basic radio functionalities rather than implementing the same functionalities in dedicated hardware chipsets (as it is done today). Whether the radio software is implemented in efficient embedded computational environment such as *digital signal processors (DSPs)* and *field programmable gate arrays (FPGAs)* or as processes running in the CPU of normal computer, the same modular approach is

used. The *physical (PHY)* layer's digital functionalities such as modulation and demodulation, frequency synthesis, and other supporting digital functions are mostly defined in code, which can be run in a computer. Figure 6.1 lower figure shows the elements that can be defined in software, which include the protocol functionality. The only hardware left out of traditional radio is its antenna (which is statically designed to cover a certain portion of the frequency spectrum), and the analog-to-digital and digital-to-analog converters, which offer the last hardware interface between the antenna and the computer designed radio. The upper picture shown in Figure 6.1 demonstrates the modularity in implementing SDR, while using the same front-end hardware (including the antenna, amplifiers, and analog-to-digital/digital-to-analog converters), and possibly modifying the same digital front end, which might involve preprogrammed FPGAs, DSPs, or ASICs.

As an example, taking two protocols functioning in the same frequency band (e.g., *Industrial, scientific, medical band (ISM-band)* 2.4 - 2.8 GHz) like IEEE 802.11g and IEEE 802.15.4 [19] (Zigbee), the two standards are very different in their modulation and MAC layers. Once these two radios are defined in software, it is possible to develop techniques to sense the spectrum for both standards simultaneously even if it is more sensible to allow a node to transmit using only one of the two. While communicating using WLAN, a Zigbee can be detected nearby, while using the same antenna, but interpreting the Zigbee beacon in software. Signal strength could be interpreted as the better channel, leading to a vertical handover. This opportunistic switch between technologies, can occur very fast. The problem is the lack of context information describing the beacon, unless more complex functionalities of the different software radio modules are turned on (such as interpreting the link layer information, and negotiate access rights). Assuming an FPGA-based implementation of the SDR, the FPGA is configured to run the WLAN protocols, whereas the other ISM band interpretation modules could be run on the CPU for flexibility. Once a handover occurs, the FPGA is reconfigured to run the Zigbee protocols. Now assuming, that context could be piggybacked in the beacon received for each signal, more modular components need to run in parallel, which then resembles the situation, where in traditional radio implementations, all network interface cards are switched on to detect all possible beacons continuously.

Although this approach is quite promising, it is unclear how the large energy footprint of SDF can be overcome. The existing proposals attempt to use SDF at base

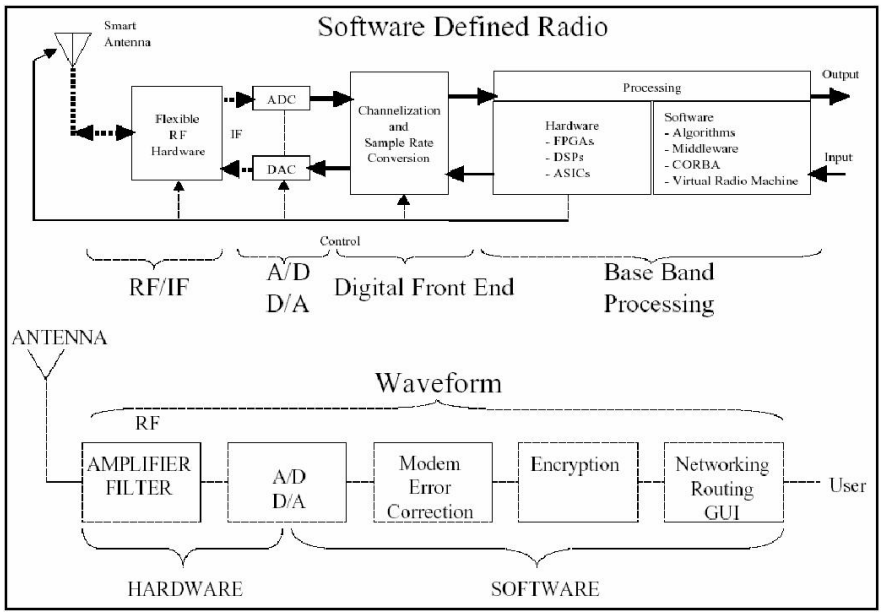

Figure 6.1: A Modular Approach to Software defined radios - source Wikipedea

station and access point level, where energy efficiency is not a problem.

6.3 Wireless Networks Evolution

The future of wireless technologies is hard to predict. The current trend is to integrate wireless techniques in industrial environments in order to enable large scale sensor networks and Internet of things. For this purpose, a further wireless connectivity service is emerging in the area of *wireless mesh networks (WMNs)* [28]. Although WMNs rely so far on self-organising mechanisms to structure large scale multi-hop networks while facing new application demands such as resilient connectivity, and isochronous communication channels (e.g., industrial WMN based on the WirelessHart technology). The latter example is used to provide low bandwidth slotted TDMA synchronous channels in large industrial WMNs. The time slots can be used to guarantee isochronous control messages to be delivered with time guarantees in a distributed environment.

6.3.1 Multihop Communication and Wireless Mesh Networks

Mesh networks refer to randomly spread wireless network nodes, that are capable of emerging into structured networks. WMNs rely on wireless links between the network

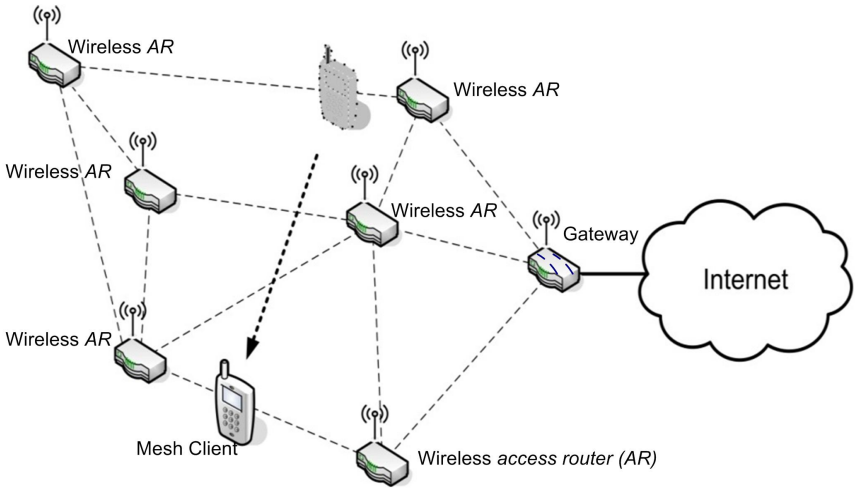

Figure 6.2: Pure Wireless Mesh Networks - Connecting a WMN through a single gateway to the remaining Internet

nodes, which take the role of routers in the WMN structure. Besides the ease of deployment (since no cables are required to connect the WMN routers), WMNs offer a cheap way to provide connectivity in hard-to-access areas, catastrophes managements, or in rapidly deployable networks.

The pure mesh (illustrated in Figure 6.2) approach assumes a single technology connecting wireless access routers forming the wireless mesh backbone infrastructure. Each static access router can maintain a long term wireless connectivity to other routers within its radio reach. The emerging structure of the network can be based on short-path algorithms like *open shortest path first (OSPF)* (which maintain connectivity within an IP-based autonomous system). Simultaneously, each access router serves several MNs within its reach, also known as mesh clients (a mesh client is shown to move within a WMN in Figure 6.2) [28].

The other architectural alternative is to assume that there is a backbone wireless infrastructure connecting the WMN routers together, whereas each router might be serving a whole wireless network such as UMTS, or satellite networks. This hybrid-architecture is assumed to organise mesh connectivity islands into interconnected macro-, mini-, and micro- cells [28].

6.3.2 Mobility in Wireless Mesh Networks

Mobility management in WMNs is still an open research issue. When taking the pure WMN architecture, the candidate solution to mobility could rely on a variant of mobile IP (e.g., hierarchical mobile IP). The way a WMN can be divided in IP autonomous systems, leads to a partition of concern. While the MN moves within the reach of a single WMN IP domain, intra-domain handovers only need to be propagated within the WMN. A single access router could be assigned to each MN entering the domain. The latter access router can then be used as an indirection point within the WMN. In that case, an intra-domain handover leads to a location update at the latter access router, similar to the *mobile anchor point (MAP)* in hierarchical mobile IP. In order to reach the MN, all traffic is redirected through the MAP, which is assumed to always keep the most up-to-date location attribute of the MN within the WMN. Another alternative to MAPs is the use of *ad-hoc on demand distance vector (AODV)* routing protocol a direct route can be used between the MN and the serving gateway. In this case, the whole route needs to be updated at each intra-domain handover.

An inter-domain handover occurs once the MN leaves the coverage range of an WMN autonomous system to another, leading to a gateway handover too. This change is dealt with through indirect routing with the help of mobile IP [30; 68]. The home agent is then informed of this type of handover. In [68], a traffic engineering approach assures that an inter-gateway handover procedure can preserve an acceptable level of QoS perceived by each user, while limiting unnecessary handovers. This problem becomes more interesting, when several gateways connect the same WMN to the Internet.

Chapter 7

Conclusions and Summary

This work is a contribution towards solving the mobility management problem in heterogeneous wireless networks, as foreseen in the fourth generation networks (4G). Although the exact list of wireless link technologies that should be integrated in the new 4G architecture is not yet fixed, one development direction should be able to support adapting connectivity to different use cases, where certain applications and services are best suited to given wireless technologies. How to make sure that certain application flows are transported on the most suitable link technology requires a new holistic approach in order to support always-best connectivity. This thesis also proposes a new way to operate and manage wireless networks the traditional boundaries of single operators, or single optimisation dimensions found in traditional mobility management architectures. This multi-dimensional optimisation problem is addressed by a context-aware approach. Context provides a way to handle the complexity added through including new optimisation dimensions. Context is also about unifying wireless, and therefore mobility architecture, heterogeneity through a semantic or meta data approach.

The problem of heterogeneity is rooted in the different architectural assumptions behind each wireless technology, which makes the case for a global mobility management hard to design in the first place. Whereas cellular networks are best suited to manage a single wireless link technology, where wireless resources are tightly controlled and allocated to mobile users, IP architecture, on the other hand, caters for interworking between heterogeneous networks. However, IP hides away link and network characteristics and is, on its own, unsuitable for advanced connectivity management and supporting always-best connectivity.

The idea of adding intelligence to mobility management in the form of context-awareness entails some deep architectural changes that have been addressed in this thesis. The benefit of semantic knowledge around the application's environmental conditions and situation is a well known research field. In this work, adding the semantics describing the underlaying networks in the form of network-centred context provides the necessary information to achieve always-best-connectivity.

The introduced overhead is necessary for collecting the context information, evaluate it, and then adapt the connectivity in the form of a context-triggered handover. In this context-management chain the work carried out in this thesis focuses on the context collection problem with the surrounding environment as it is the most considerable communication overhead added by this architecture. The other two aspects of context management are also addressed to complement the evaluation of hypothetical system. It is also clear that the adoption of such a system needs to be carried out closely within some standardisation efforts such as those done in IEEE 802.21 media independent handover (See Section 6.2 in Chapter 6).

7.1 Main Contributions of the Thesis

The context collection process has been designed in this thesis to suit a dynamic and mobile environment. The task is formulated as a distributed data management problem in a mobile environment in Chapter 3. Network related context is generated by a common abstraction layer describing each of the heterogeneous systems through semantic information (or meta data). The context information hides the complexity of each wireless system, but still can reflect the dynamic status of the wireless resources. While assuming that each wireless resource can be tracked by its native management system, the status and characteristics of the wireless resource can be deposited on some local gateway node. It is, then, necessary to translate between the local management information and the context model used to describe single network entities and their dynamic status. The semantic boundary of each context entity is drawn on top of the domain boundary covered by each management entity. Since the sources of the network context are scattered in nature, the problem becomes that of collecting dynamic context and making it available for the mobile user in a continuous manner. The latter constraints exclude a centralised system or even a tier-based hierarchical system like

DNS, since the dynamic nature of both stored content and the request load is much higher than in DNS.

Instead, the use of overlays is advocated in this thesis, and more specifically semantic overlays, to manage context in a mobile environment. The choice of overlays is due to their ability to distribute request and query load between all parts of the application-level network. The structure of the network can be manipulated in order to optimise several aspects of the application behaviour simultaneously. One structuring aspects results from the user behaviour and need for continuous and spatio-temporal queries. The other aspect relates to how to interconnect network management entities that sense and collect network context. The last structuring aspect ralates to how to cluster data to avoid bottlenecks in the request process and enable load balancing or load distribution and aggregation of similar requests.

The resulting architecture first places a proxy comparable to both a series of home agents and HLR/VLR brought together. The proxy's role is to take over the collection of network context for the mobile user, while adapting this collection process to the user's context changes. The proxy nodes, named search peers, are part of an overlay comparable with HLR/VLR in cellular networks, and therefore, could support call delivery by implementing a paging mechanism. The search peers face similar scalability requirements as the cellular network assumptions on user arrival rates at VLRs. They also cache similar static network information, which can be retrieved by users with aggregate movement behaviour. The exact design of caching and user aggregate behaviour strategies are out of the scope of this work, and therefore, they are only discussed in the scope of some related studies on the field of Mobile P2P solutions and traditional mobility management work.

7.2 Development Work and Followed Methodology

The context management constraints, mentioned above, are used in the design process of a semantic overlay in Chapters 3 and 4. It is shown that the most suitable clustering criteria for the data describing wireless networks is their geographic attribute. The geo-clustering can be generalised to any other mobile location-based services, where continuous and spatio-temporal queries need to be supported by separate LBS providers. The overlay connects management entities located within some domain boundaries like

operator and technology boundaries, and re-organises the information according to its concentration in a location-based manner. The implemented framework is explained and analysed in Chapter 4. This geographic clustering method is optimised to suit one of the most efficient overlay routing protocols, namely Chord.

The mapping used between the overlay protocol layer and and the data models, consists the novelty of this work. A space-filling curve allows to describe 2-dimensional space as a fractal, whose depth reflects the object density in any subspace (seen as the number of objects or data items within a certain subspace or area). The density of service providers seen as the number of nodes joining the system is also taken into account when structuring the overlay. The fractal structure captures the density of data items or objects in a certain part of the space (reflecting the difference between urban and rural subspaces, for instance). The algorithm relies on the DHT protocol to discover this spatial density. The fractal space model results into a binary prefix relationship, which can also be used to construct multiple level hierarchical CHORD rings, where each ring shares the same prefix length. The self-organising structuring algorithm effects directly the associated query (or context collection) effort. It can also limit the diameter of each node in the overlay and, therefore, influences the associated overlay maintenance cost.

The evaluation methodology of the space-model has been to compare analytically the effect of each design parameter and each property of the space-filling curve on the complexity of the communication effort when performing spatial range queries. The analytic results are then validated by simulation to demonstrate the possibility to design the system parameters quite accurately, in advance.

Further to the data management system, the integration of the context reasoning or connectivity triggering have been addressed in this thesis. It has been shown that it is possible to either integrate this process into an existing framework, like IEEE 802.21, or to implement this separately, based on an purely application-layer decision process. The simulative implementation of the framework and its evaluation are given in Chapter 5. The application layer reasoning and context-based decision process has been designed to interact with lower layers (i.e., link and network layers) through technology hooks. The latter hooks are proprietary to each link technology. The standardisation efforts made within the IEEE 802.21 focus on the development of these interactions. IEEE 802.21, however, only tracks the status of each wireless link, while not yet integrating other

type of technology context, such as the available network resources (such as bandwidth on the wireline part of the network), security, etc. A more holistic approach beyond the last mile's wireless quality, is still lacking. The network context is multi-dimensional descriptive meta-data, which can still encapsulating the methods and interaction processes that specify the way to interact with the lower layers.

7.3 Future Work

The standardisation effort required to integrate context-awareness in 4G networks can be facilitated by the full use of extendible ontologies to further integrate different functions of different wireless technologies or complementary embedded functionalities (such as an adaptive location tracking) in the context model.

The use of W3C technologies such as web ontology language (OWL) or resource description framework (RDF) should be part of in the standardisation efforts to define minimum templates and technology specific description schemas. This can be defined in inter-operability standards such as IEEE 802.21. The advantage of such an approach is that it still allows some room for extending the ontologies to include additional functions specific to certain manufacturers.

On the validation front, it is valuable to compare the analytic model with an actual real decentralised LBS P2P system. Such a real implementation could, for example, be through extending the existing FON management system to other heterogeneous networks, while supporting real-time or dynamic tracking of network resources.

Another issue that still has to be dealt with is developing the right business model, to encourage cooperation between network providers. Roaming between operator boundaries or technologies is still lacking transparent or flexible payment models. Instead, the operators currently charge the user exorbitant amounts of money for using their license-free WLAN networks, while trying to reduce cost on their licensed cellular spectrum. The FON business model could offer a good alternative to private users sharing their WLAN. The only paying users in the FON network are foreign users that pay for access, whereas users that have shared their DSL access get free access to the whole FON network. Similar cooperative business models should be investigated with the goal of sharing income between operators, while providing enough incentives to roll out new infrastructure.

Security is also another complex issue that needs to be addressed explicitly. The assumption made in this thesis is that security can be modelled one of the context dimensions. At the network level, each network has to be classified according to some trust and security metric, which differentiates between secure networks and less secure ones. Depending on the user activity and profile, networks can be selected. The selection of heterogeneous security and trust levels can match the user activity. For instance, if the user is currently accessing his/her company's network, the selection process requires a more secure wireless connection to when the same user is just reading the news. Here an open WLAN in the local cafe can be selected automatically.

Bibliography

[1] Press Release, Geneva, 2 March 2009 - International Telecommunication Union (last checked in March 2009) - URL: http://www.itu.int/newsroom/press_releases/2009/07.html. 1

[2] IEEE 802.11 Working Group (last checked in November 2008), URL: http://standards.ieee.org/getieee802/802.11.html. 2, 135, 136, 137

[3] European Telecommunications Standards Institute, (Last checked in November 2008), URL: http://www.etsi.org/. 20

[4] 3rd Generation Partnarship Project (last checked on November 2008), URL: http://www.3gpp.org/. 20

[5] High-Speed Downlink Packet Access (HSDPA), Wikepedia (last checked in September 2008), URL: http://en.wikipedia.org/wiki/High-Speed_Downlink_Packet_Access. 20

[6] Fourth Generation (4G), Wikepedia (last checked in September 2008), URL: http://en.wikipedia.org/wiki/4G. 20

[7] 3GPP, Long Term Evolution of the 3GPP radio technology (last checked in December 2008), URL: http://www.3gpp.org/Highlights/LTE/LTE.htm. 21

[8] Daidalos EU Project: Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services, (Last checked in October 2008), URL: http://www.ist-daidalos.org/. 22, 29

[9] IEEE 802.21 Working Group (last checked in November 2008), URL: http://www.ieee802.org/21/. 29

[10] The Placelab (last checked in December 2008), URL:http://www.placelab.org. 62, 169

[11] Elvin: "A system for content-based notification routing", (last checked in November 2008), URL: http://elvin.org/. 67

[12] Napster (last checked in November 2008), URL: http://www.napster.com/. 67

[13] What's FON, (last checked on December 2008), URL: http://www.fon.com/en/info/whatsFon. 67

[14] Earth Geography, Wikepedia (last checked November 2008), URL: http://en.wikipedia.org/wiki/Earth. 98

[15] The Network Simulator - ns-2 (last checked in December 2008), URL:http://www.isi.edu/nsnam/ns/. 152

[16] OPNET Modeler tool (last checked in December 2008), URL: http://www.opnet.com. 152

[17] OMNeT++ IPv6Suite (last checked in December 2008), URL:http://ctieware.eng.monash.edu.au/twiki/bin/view/Simulation/IPv6Suite. 152

[18] The official Software Defined Radios Forum (last checked in December 2008), URL: http://www.sdrforum.org/. 174

[19] IEEE 802.15 WPAN Task Group 4 (TG4) (last checked in April 2009), URL: http://www.ieee802.org/15/pub/TG4.html. 175

[20] *GSM 03.60, GPRS, Service description, Stage 2.* 23

[21] **MobiWan: NS-2 extensions to study mobility in Wide-Area IPv6 Networks**. http://www.inrialpes.fr/planete/mobiwan/, 2004. MOTOROLA Labs Paris in collaboration with INRIA PLANETE Team. 152

[22] **MPHPT considers GPS is a must have feature of future 3G phones**. http://www.3gnewsroom.com/phorum-3.4.8a/read.php?f=1\&i=2033\&t=2033, 2004. 3

[23] K. Aberer, A. Datta, and M. Hauswirth. **The Quest for Balancing Peer Load in Structured Peer-to-Peer Systems**. Technical Report, Ecole Polytechnique Federale de Lausanne, LSIR-REPORT-2003-003, 2003. 133

[24] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. **Towards a Better Understanding of Context and Context-Awareness**. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307, London, UK, 1999. Springer-Verlag. 47, 53

[25] N. Abramson. **The ALOHA System - Another Alternative for Computer Communications**. *AFIPS Conference Proceedings*, **Vol39**:295–298, 1970. 1

[26] R. Aguiar, P. Antoniadis, J. P. Barraca, L. Berlemann, M. D. de Amorim, A. Duda, S. Fdida, P. Fernandes, L. Iannone, C. Ibars, J. Mangues-Bafalluy, R. Rocha, U. Röthlisberger, F. Rousseau, A. Satsiou, C. Schwingenschlögl, L. S. F., L. Tassiulas, and A. Ziller. **Deliverable D1.2 - Architectural requirements for the Radio Internet: addressing, routing, design strategies**. EU Project Deliverable, WIP Grant Number 27402 ISTŰ4027402-WIP-D1.2, EU Project WIP, October 2006. 2

[27] T. Ahmed, K. Kyamakya, and M. Ludwig. **Architecture of a Context-Aware Vertical Handover Decision Model and Its Performance Analysis for GPRS - WiFi Handover**. In *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 795–801, Washington, DC, USA, 2006. IEEE Computer Society. 61

[28] I. F. Akyildiz, X. Wang, and W. Wang. **Wireless mesh networks: a survey**. *Comput. Netw. ISDN Syst.*, **Vol.47**(4):445–487, 2005. 50, 176, 177

[29] I.F. Akyildiz, Jiang Xie, and S. Mohanty. **A survey of mobility management in next-generation all-IP-based wireless systems**. *Wireless Communications, [see also IEEE Personal Communications]*, **Vol.11**(4):16–28, August 2004. 9, 14, 18, 19, 40

[30] Y. Amir, C. Danilov, M. Hilsdale, R. Musăloiu-Elefteri, and N. Rivera. **Fast handoff for seamless wireless mesh networks**. In *MobiSys*

'06: Proceedings of the 4th international conference on Mobile systems, applications and services, pages 83–95, New York, NY, USA, 2006. ACM. 178

[31] F. Andersen, H. De Meer, I. Dedinski, C. Kappler, A. Mäder, J. Oberender, and K. Tutschku. **An Architecture Concept for Mobile P2P File Sharing Services**. In *Workshop at Informatik 2004 - Algorithms and Protocols for Efficient Peer-to-Peer Applications*, pages 229–233, Ulm, 9 2004. 80

[32] O. Angin, A. T. Campbell, M. E. Kounavis, and R. R. f. Liao. **Open programmable mobile networks**. In *Proc. Eight Intl Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV*, 1998. 56

[33] S. Aust, D. Proetel, N. A. Fikouras, C. Görg, C. Pampu, and Siemens Ag. **Policy based Mobile IP handoff decision (POLIMAND) using generic link layer information**. In *In 5th IEEE International Conference on Mobile and Wireless Communication Networks (MWCN)*, 2003. 60

[34] A.Varga and R. Hornig. **An overview of the OMNeT++ simulation environment**. In *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 151

[35] A.-E. Baert and D. Seme. **Voronoi mobile cellular networks: topological properties**. *Parallel and Distributed Computing, 2004. Third International Symposium on/Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks, 2004. Third International Workshop on*, pages 29–35, July 2004. 148

[36] F. Bai, Narayanan Sadagopan, and A. Helmy. **IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks**. In *INFOCOM 2003: Proceedings*

of the twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies, Vol **Vol.2**, pages 825–835. IEEE, 30 March-3 April 2003. 118

[37] S. Balasubramaniam and J. Indulska. **Vertical handover supporting pervasive computing in future wireless networks**. *Computer Communications*, **Vol.27**:708–719, may 2004. 61

[38] M. Baldauf, S. Dustdar, and F. Rosenberg. **A survey on context-aware systems**. *IJAHUC*, **Vol.2**(4):263–277, 2007. 46, 51

[39] P. Bellavista, A. Corradi, and L. Foschini. **Context-aware handoff middleware for transparent service continuity in wireless networks**. *Pervasive Mob. Comput.*, **Vol.3**(4):439–466, 2007. 3, 4

[40] P. Bellavista, A. Corradi, and C. Giannelli. **A layered infrastructure for mobility-aware best connectivity in the heterogeneous wireless internet**. In *MOBILWARE '08: Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, pages 1–8, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 3

[41] C. Bettstetter, H.-J. Voegel, and J. Eberspraecher. **GSM Phase 2+: General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface**. *IEEE Communications Surveys Ţ http://www.comsoc.org/pubs/surveys Ţ Third Quarter*, **vol. 2**(3), 1999. 20, 23, 25

[42] A. R. Bharambe, M. Agrawal, and S. Seshan. **Mercury: supporting scalable multi-attribute range queries**. In *SIGCOMM 2004: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 353–366, New York, USA, 2004. ACM Press. 89

[43] E. Bortnikov, I. Cidon, and I. Keidar. **Scalable real-time gateway assignment in mobile mesh networks**. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM. 66

[44] C. Bouras, Y. Siahos, and P. Spirakis. **An efficient algorithm to improve handoff in mobility for IPv6 protocol**. *8th International Conference on Software, Telecommunications and Computer Networks-SoftCOM 2000*, **Vol.1**:505–511, 2000. 143

[45] M. Buddhikot, G. Chandranmenon, S. Han, Y.W. Lee, S. Miller, and L. Salgarelli. **Integration of 802.11 and third-generation wireless data networks**. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, **vol.1**:503–512, March-3 April 2003. 2, 28

[46] A. Campbell, J. Gomez, C-Y. Wan, Z. Turanyi, and A. Valko. **Cellular IP,**. draft-valko-cellularip-01.txt, Internet Draft, IETF, October 1999. (Work in progress). 40

[47] A. T. Campbell, M. E. Kounavis, and R. R. f. Liao. **Programmable Mobile Networks**. *Computer Networks and ISDN Systems*, **Vol.31**:741–765, 1999. 56

[48] A.T. Campbell, J. Gomez, S. Kim, A.G. Valko, Chieh-Yih Wan, and Z.R. Turanyi. **Design, implementation, and evaluation of cellular IP**. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, **Vol.7**(4):42–49, Aug 2000. 40

[49] R. Chakravorty, P. Vidales, L. Patanapongpibul, K. Subramanian, I. Pratt, and J. Crowcroft. **On Internetwork Handover Performance using Mobile IPv6**. Technical report, University of Cambridge Computer Laboratory, June 2003. 143

[50] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, and J. Hellerstein. **A case study in building layered DHT applications**. In *SIGCOMM 2005: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 97–108, New York, USA, August 22Ű26 2005. ACM Press. 91

[51] G. Chen and D. Kotz. **A Survey of Context-Aware Mobile Computing Research**. Technical Report TR2000-831, Dartmouth College US, 2000. 47, 51

[52] T. Cornall, B. Pentland, and P. Khee. **Improved Handover Performance in Wireless Mobile IPv6**. *The 8th International Conference on Communication Systems*, **Vol.2**:857–861, 2002. 143, 159

[53] J. Cowling. **Dynamic Location Management in Heterogeneous Cellular Networks**, November 2004. Bachelor Thesis, University of Sydney, Australia. 9, 197

[54] A. Crespo and H. Garcia-Molina. **Semantic Overlay Networks for P2P Systems**. In Gianluca Moro, Sonia Bergamaschi, and Karl Aberer, editors, *Agents and Peer-to-Peer Computing, Third International Workshop, AP2PC 2004, New York, NY, USA, July 19, 2004, Revised and Invited Papers*, Vol **3601** of *LNCS*, pages 1–13. Springer, 2005. 72

[55] F. van Harmelen D. L. McGuinness. **OWL Web Ontology Language Overview**, 10 February 2004. W3C Recommendation (last checked in December 2008), URL:`http://www.w3.org/TR/owl-features/`. 52

[56] A. Datta. *SOS: Self-Organizing Substrates*. PhD thesis, École Polytechnique Fédérale De Lausanne, 2006. 91, 92

[57] A. Datta, M. Hauswirth, R. John, R. Schmidt, and K. Aberer. **Range Queries in Trie-Structured Overlays**. In *P2P '05: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing*, pages 57–66, Washington, DC, USA, 2005. IEEE Computer Society. 90, 91, 92

[58] S. Deering. **ICMP Router Discovery Messages**. Internet Engineering Task Force: RFC 1256, September 1991. 35

[59] N. Van den Wijngaert and C. Blondia. **A Location Augmented Low Latency Handoff Scheme for Mobile IP**. In *In Proc. of the 1st Conference on Mobile Computing and Ubiquitous Networking (ICMU04)*, pages 180–185, January 2004. 143

[60] S. G. Denazis, K. Miki, J. B. Vicente, and A. T. Campbell. **Designing Interfaces for Open Programmable Routers**. In *IWAN '99: Proceedings of the First International Working Conference on Active Networks*, pages 13–24, London, UK, 1999. Springer-Verlag. 56

[61] J. M. Dennis. **Partitioning with Space-Filling Curves on the Cubed-Sphere**. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 269.1, Washington, DC, USA, 2003. IEEE Computer Society. 98

[62] A. K. Dey and G. D. Abowd. **Towards a Better Understanding of context and context-awareness**. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, July 1999. 47, 48, 53

[63] R. Droms. **DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)**. Internet Engineering Task Force: RFC 3646, December 2003. 142

[64] C. Du Mouza, W. Litwin, and P. Rigaux. **SD-Rtree: A Scalable Distributed Rtree**. *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 296–305, April 2007. 89

[65] A. Dutta, S. Madhani, and W. Chen. **GPS Assisted Fast-Handoff Mechanism for Real-Time Communication**. *Sarnoff Symposium, 2006 IEEE*, pages 1– 4, March 2006. 169

[66] C. Perkins E. Fogelstroem, A. Jonsson. **RFC4857 - Mobile IPv4 Regional Registration**. IETF RFC, June 2007. 39, 40

[67] L. Eastwood, S. Migaldi, Qiaobing Xie, and V. Gupta. **Mobility using IEEE 802.21 in a heterogeneous IEEE 802.16/802.11-based, IMT-advanced (4g) network**. *Wireless Communications, IEEE*, **Vol.15**(2):26–34, April 2008. xii, 21, 22, 29, 30, 31

[68] E.Bortnikov, I. Cidon, I. Keidar, T. Kol, and A. Vaisman. **A QoS WMN with mobility support**. *SIGMOBILE Mob. Comput. Commun. Rev.*, **Vol.12**(1):46–48, 2008. 66, 178

[69] N. Efthymiou, Yim Fun Hu, and R.E. Sheriff. **Performance of inter-segment handover protocols in an integrated space/terrestrial-UMTS environment**. *Vehicular Technology, IEEE Transactions on*, **Vol.47**(4):1179–1199, Nov 1998. 28

[70] M. Egen, S. Coleri, B. Dundar, J. Rahul, A. Puri, and P. Varaiya. **Application of GPS to Mobile IP and routing in wireless networks**. *Proceedings of the 56th Vehicular Technology Conference*, **Vol.2**:1115–1119, 2002. 143

[71] E. Ekici and C. Ersoy. **Multi-Tier Cellular Network Dimensioning**. *Wirel. Netw.*, **Vol.7**(4):401–411, 2001. 19

[72] M. Graf Esterhazy. **Location Aided Handover in MIPv6**. Diploma thesis under the supervision of Amine Houyou, 2005. 148, 161

[73] C. Fan, M. Schlager, A. Udugama, V. Pangboonyanon, A. C. Toker, and G. Coskun. **Managing Heterogeneous Access Networks Coordinated policy based decision engines for mobility management**. In *LCN '07: Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pages 651–660, Washington, DC, USA, 2007. IEEE Computer Society. 60

[74] F. Feng and D. S. Reeves. **Explicit Proactive Handoff with Motion Prediction for Mobile IP**. **Vol.2**:855–860, March 2004. 143

[75] A. Ferscha, C. Holzmann, and S. Oppl. **Context awareness for group interaction support**. In *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 88–97, New York, NY, USA, 2004. ACM. 46

[76] J. Gao. *A Distributed and Scalable Peer-to-Peer Content Discovery System Supporting Complex Queries.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 2004. 68, 69

[77] R. Giaffreda, A. Karmouch, A. Jonsson, A.M. Kalrsson, M.I. Smirnov, R. Glitho, and A. Galis. **Context-aware Communication in Ambient Networks**. Wwrf 11, Oslo, Norway, Jun 10-11, 2004. 49

[78] V. Casares Giner. **State of the art in Location Management procedures**. EuroNGI Deliverable D.JRA.1.5.1, 2004. 9, 11

[79] H. Grine, T. Delot, and S. Lecomte. **Adaptive query processing in mobile environment**. In *MPAC '05: Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*, pages 1–8, New York, NY, USA, 2005. ACM. 64

[80] W. Guan, X. Ling, X. (Sherman) Shen, and D. Zhao. **Handoff trigger table for integrated 3G/WLAN networks**. In *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 575–580, New York, NY, USA, 2006. ACM. 28

[81] E. Gustafsson and A. Johnson. **Always Best Connected**. **Vol.10**:49–55, February 2003. 3

[82] Y. Gwon, G. Fu, and R. Jain. **Fast handoffs in wireless LAN networks using mobile initiated tunneling handoff protocol for IPv4 (MITHv4)**. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, **Vol.2**(vol.2):1248–1253, March 2003. 41

[83] H. Haverinen and J. Malinen. **Mobile IP Regional Paging**. Internet Draft, IETF, June 2000. (Work in progress). 40

[84] D. Hernando and V. Crespi. **Sampling theory for process detection with applications to surveillance and tracking**. Vol **Vol.5403**, pages 143–152. SPIE, 2004. 13

[85] J. Hightower, A. LaMarca, and I.E. Smith. **Practical Lessons from Place Lab**. *Pervasive Computing, IEEE*, **Vol.5**(3):32–39, July-Sept. 2006. 91, 169

[86] H. Honkasalo, K. Pehkonen, M.T. Niemi, and A.T. Leino. **WCDMA and WLAN for 3G and beyond**. *Wireless Communications, IEEE*, **Vol.9**(2):14–18, April 2002. 28

[87] J. Hou and D.C. O'Brien. **Vertical handover-decision-making algorithm using fuzzy logic for the integrated Radio-and-OW system**. *Wireless Communications, IEEE Transactions on*, **Vol.5**(1):176–185, Jan. 2006. 59

[88] A. M. Houyou, H. de Meer, and M. Esterhazy. **P2P-based Mobility Management for Heterogeneous Wireless Networks and Mesh Networks**. In L. Fratta M. Cesana, editor, *Proceedings of EuroNGI Joint Workshops IA 8.3 and IA 8.2 LNCS 3883*, Vol **LNCS**. Springer Verlag, July 2005. 3

[89] A. M. Houyou and H. De Meer. **Self-Organizing Location-Aware Overlay Networks**. In David Hutchison and Randy H. Katz, editors, *IWSOS 2007: Proceedings of the 2nd International Workshop on Self-Organizing Systems*, The Lake District, UK, September 11-13 2007. 3, 55

[90] A. M. Houyou and H. De Meer. *Lecture Notes in Geoinformation and Cartography*, chapter Efficient Overlay Mediation for Mobile Location-Based Services, pages 353–371. Springer, Berlin Heidelberg, October 2008. 114

[91] A. M. Houyou, A. Stenzer, and H. De Meer. **Performance Evaluation of Overlay-based Range Queries for Mobile Systems**. In L. Cerdà-Alabern, editor, *Wireless Systems and Mobility in Next Generation Internet, 4th International Workshop of the EuroNGI/EuroFGI Network of Excellence, Barcelona, Spain, January 16-18, 2008, Revised Selected Papers*, Vol **5122** of *LNCS*, pages 201–219. Springer, 2008. 4, 55

[92] R. Hsieh, Z.-G. Zhou, and A. Seneviratne. **S-MIP: A seamless handoff architecture for Mobile IP**. *Proceedings of INFOCOM 2003*, 2003. 143

[93] N. T. T. Huong and M. Mitsuji. **Ontology-Based Context Management Agent for Vertical Handoff Use Fuzzy Logic Decision in Heterogeneous Network**. *Complex, Intelligent and Software Intensive Systems, 2007. CISIS 2007. First International Conference on*, pages 215–220, April 2007. 53, 54, 59

[94] IETF. **Mobility for IP: Performance, Signaling and Handoff Optimization (mipshop)**. `http://www.ietf.org/html.charters/mipshop-charter.html`, August 2008. Last modified on 21st August 2008. 2, 31

[95] Infineon. **Hammerhead - PMB 2520 Single Chip Assisted-GPS Solution**, 2005. 3, 169

[96] H. V. Jagadish. **Linear clustering of objects with multiple attributes**. In *SIGMOD '90: Proceedings of the 1990 ACM SIGMOD international conference on Management of data, Atlantic City, New Jersey, United States*, pages 332–342, New York, NY, USA, 1990. ACM. 87

[97] H. V. Jagadish, B. C. Ooi, and Q. H. Vu. **BATON: a balanced tree structure for peer-to-peer networks**. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 661–672. VLDB Endowment, 2005. 89

[98] D. Johnson, C. Perkins, and J. Arkko. **Mobility support in IPv6, RFC 3775**. Technical report, IETF, June 2004. RFC 3775. 2, 135, 141, 142

[99] M. Kan. **Data Management in Mobile P2P Systems**. Technical Report 2005-23, Stanford InfoLab, September 2005. 80

[100] M. Kassar, B. Kervella, and G. Pujolle. **An overview of vertical handover decision strategies in heterogeneous wireless networks**. *Computer Communication*, **Vol.31**(10):2607–2620, 2008. 59

[101] R. Keller, L. Ruf, A. Guindehi, and B. Plattner. **PromethOS: A Dynamically Extensible Router Architecture Supporting Explicit Routing**. In *IWAN '02: Proceedings of the IFIP-TC6 4th International Working Conference on Active Networks*, pages 20–31, London, UK, 2002. Springer-Verlag. 58

[102] J. Kempf, J. Wood, and G. Fu. **Fast mobile IPv6 handover packet loss performance: measurement for emulated real time traffic**. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, **Vol.2**:1230–1235, March 2003. 40, 41, 143

[103] S. Kent and R. Atkinson. **IP Encapsulating Security Payload (ESP)**. Internet Engineering Task Force: RFC 2406, November 1998. 143

[104] T. Keshav. **Location Management in Wireless Cellular Networks**. Technical report, Technical Report at Washington University of St. Louis, 2006. 10

[105] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. **Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0**. W3C Recommendation, http://www.w3.org/TR/CCPP-struct-vocab/, 15 January 2004. 52, 53, 57

[106] M. Knoll and T. Weis. **Optimizing Locality for Self-organizing Context-Based Systems**. In Hermann de Meer and James P. G. Sterbenz, editors, *IWSOS 2006: Proceedings of the 1st International Workshop on Self-Organizing Systems*, Vol **4124** of *LNCS*, pages 62–73, Berlin Heidelberg, 2006. Springer. 101

[107] N. E. Kolodziej. **Location management in cellular networks using reliable metrics**. Cited by [53], 2003. 12

[108] R. Koodli. **Fast Handovers for Mobile IPv6**. Rfc 4068, IETF, July 2005. 2, 41

[109] M. E. Kounavis, A. T. Campbell, G. Ito, and G. Bianchi. **Design, implementation, and evaluation of programmable handoff in mobile networks**. *Mob. Netw. Appl.*, **Vol.6**(5):443–461, 2001. 27, 56

[110] M.E. Kounavis, A.T. Campbell, G. Ito, and G. Bianchi. **Supporting programmable handoff in mobile networks**. *Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on*, pages 131–141, 1999. 56

[111] D. Kouvatsos, S. Adli Assi, I. Mkwawa, V. Casares Giner, H. De Meer, and Amine Houyou. *An Information Theoretic Approach to Mobility Management: An Overview*, pages 10 – 25. July 2005. 53, 55

[112] W. Ku, R. Zimmermann, and H. Wang. **Location-Based Spatial Query Processing with Data Sharing in Wireless Broadcast Environments**. *Mobile Computing, IEEE Transactions on*, **Vol.7**(6):778–791, June 2008. 81

[113] J. Lai, E. Wu, A. Varga, A. Sekercioglu, and G. Egan. **A Simulation Suite for Accurate Modeling of IPv6 Protocols**. *Proceedings of the 2nd international OMNeT++ Workshop*, **Vol.2**:2–12, 2002. 152, 153, 154

[114] L. Lan and H. Wen-Jing. **Localized delaunay triangulation for topological construction and routing on manets**. In *Proceedings of 2nd ACM Workshop on Principles of Mobile Computing (POMC'02)*, 2002. 148

[115] B. Lane. **Cognitive radio technologies in the commercial arena**. In *Proceedings of FCC Workshop on Cognitive Radios*, May 19 2003. 3

[116] C. W. Lee, L. M. Chen, M. C. Chen, and Y. S. Sun. **A framework of handoffs in wireless overlay networks based on mobile IPv6**. *Selected Areas in Communications, IEEE Journal on*, **Vol.23**(11):2118–2128, Nov. 2005. 28, 143

[117] R. Li, J. Li, K. Wu, Y. Xiao, and J. Xie. **An Enhanced Fast Handover with Low Latency for Mobile IPv6**. *Wireless Communications, IEEE Transactions on*, **Vol.7**(1):334–342, Jan. 2008. 40, 41, 142, 143

[118] T. Liu, P. Bahl, and I. Chlamtac. **Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks**. *Selected Areas in Communications, IEEE Journal on*, **Vol.16**(6):922–936, Aug 1998. 41, 171, 172

[119] A. Garcias-Martinez M. Bagnulo, I. Soto and A. Azcorra. **Avoiding DAD for Improving Real-Time Communication in MIPv6 Environments**. **2515**:73–79, 2002. 142, 155

[120] J. Manner and M. Kojo. **Mobility Related Terminology**. IETF: http://www.isi.edu/in-notes/rfc3753.txt, June 2004. 4

[121] B. Mathieu, Meng Song, M. Brunner, M. Stiemerling, M. Cassim, A. Galis, L. Cheng, K. Jean, R. Ocampo, Zhaohong Lai, and M. Kampmann. **Autonomic Management of Context-Aware Ambient Overlay Networks**. *Communications and Networking in China, 2007. CHINACOM '07. Second International Conference on*, pages 727–733, Aug. 2007. 49

[122] P. Maymounkov and D. Mazi. **Kademlia: A Peer-to-Peer Information System Based on the XOR Metric**. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 53–65, London, UK, 2002. Springer-Verlag. 90

[123] M. McHenry. **Frequency Agile Spectrum Access Technologies**. In *FCC Workshop on Cognitive Radios*, May 19 2003. 3

[124] J. McNair, I.F. Akyildiz, and M.D. Bender. **An inter-system handoff technique for the IMT-2000 system**. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, **Vol. 1**:208–216, 2000. 28

[125] H. De Meer. **Lecture Series summer term 2005: Peer-to-Peer Networking**. Universität Passau, 2005. 42

[126] T. Melia, A. de la Oliva, I. Soto, C. J. Bernardos, and A. Vidal. **WLC34-2: Analysis of the Effect of Mobile Terminal Speed on WLAN/3G Vertical Handovers**. In *Global Telecommunications Conference, Nov. 2006. GLOBECOM '06. IEEE*, pages 1–6, 2006. 3

[127] R. Min and A. Chandrakasan. **MobiCom poster: top five myths about the energy consumption of wireless communication**. *SIGMOBILE Mob. Comput. Commun. Rev.*, **Vol.7**(1):65–67, 2003. 174

[128] A. Misra, S. Das, A. Dutta, A. McAuley, and S.K. Das. **IDMP-based fast handoffs and paging in IP-based 4G mobile networks**. *Communications Magazine, IEEE*, **Vol.40**(3):138–145, Mar 2002. 39, 40

[129] J. Mitola. **Software Radio Architecture: Object- Oriented Approaches to Wireless System Engineering**, 2000. 4

[130] B. Moon, H.V. Jagadish, C. Faloutsos, and J. H. Saltz. **Analysis of the clustering properties of the Hilbert space-filling curve**. *IEEE Transactions on knowledge and data engineering*, **Vol.13**(1):124–141, January/February 2001. 97, 101

[131] C. Moon, S. Yang, and I. Yeom. **Performance Analysis of Decentralized RAN (Radio Access Network) Discovery Schemes for IEEE 802.21**. *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 41–45, 30 2007-Oct. 3 2007. 32

[132] E.D.S. Moreira, D.N. Cottingham, J. Crowcroft, Pan Hui, G.E. Mapp, and R.M.P. Vanni. **Exploiting contextual handover information for versatile services in NGN environments**. *Digital Information Management, 2007. ICDIM '07. 2nd International Conference on*, **Vol.1**:506–512, Oct. 2007. 55

[133] G. M. Morton. **A computer Oriented Geodetic Data Base; and a New Technique in File Sequencing**. Technical report, Technical Report, Ottawa, Canada: IBM Ltd, 1966. 87

[134] S. Kucukoner N. Ekiz, T. Salih and K. Fidanboylu. **An Overview of Handoff Techniques in Cellular Networks**. *International Journal of Information Technology*, **Vol.2**:132 – 136, 2005. 16, 17

[135] T. Narten, E. Nordmark, and W. Simpson. **RFC2461: Neighbor Discovery for IP Version 6**, 1998. 36, 38, 142

[136] T. Narten and S. Thomson. **IPv6 Stateless Address Autoconfiguration**. Internet Engineering Task Force: RFC 2462, December 1998. 142

[137] R. Ocampo. *Understanding, Modeling and Using Flow Context*. PhD thesis, University College London, 2006. 49, 51

[138] J. Pascoe. **Adding generic contextual capabilities to wearable computers**. *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 92–99, Oct 1998. 47

[139] G Peano. **Sur une courbe qui remplit toute une aire plane**. *Mathematische Annalen, Springer Berlin/Heidelberg*, **Vol.36**(1):157–160, March 1890. 88, 97

[140] C.E. Perkins. **Mobile IP**. *IEEE Communications Magazine*, **Vol.40**(5):66–82, May 2002. 2

[141] C. Pils, I. Roussaki, T. Pfeifer, N. Liampotis, and N. Kalatzis. **Federation and Sharing in the Context Marketplace**. *J. Hightower, B. Schiele, and T. Strang (Eds.): LoCA 2007*, **4718**:121Ű138, 2007. 81

[142] C. Greg Plaxton, Rajmohan Rajaraman, and Andréa W. Richa. **Accessing nearby copies of replicated objects in a distributed environment**. In *SPAA '97: Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures*, pages 311–320, New York, NY, USA, 1997. ACM. 93

[143] D. Plummer. **An Ethernet Address Resolution Protocol**. Internet Engineering Task Force: RFC 826, November 1982. 35

[144] J. Postel. **Internet Control Message Protocol**. Internet Engineering Task Force: RFC 792, September 1981. 35

[145] C. Prehofer. *Towards 4G Technologies*, chapter Context-aware Mobility Management, pages 211–234. Copyright ľ 2008 John Wiley & Sons, Ltd, 2008. xii, 48, 56, 58, 62

[146] C. Prehofer, J. Hillebrand, P. Hofmann, P. Mendes, Q. Wei, and C. Bettstetter. **Active IP Networking: Towards Self-Organized Ambient Communication**. *NTT DoCoMo Technical Journal*, **vol. 6**(1), June 2004. 53, 56, 58

[147] C. Prehofer, N. Nafisi, and Q. Wei. **A framework for context-aware handover decisions**. *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, **Vol.3**:2794–2798, Sept. 2003. 3, 4, 53, 56, 57, 58

[148] D. Preuveneers and Y. Berbers. **Towards context-aware and resource-driven self-adaptation for mobile handheld applications**. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 1165–1170, New York, NY, USA, 2007. ACM. 174

[149] C. Prehofer Q. Wei. **Context Management in Mobile Environments**. In *3rd International IST-Anwire Workshop on Adaptable Service Provision, held in conjunction with the DAIS-FMOODS conferences - Paris, 18 November*, 2003. 56, 58

[150] K. Wehrle (Eds.) R. Steinmetz, editor. *Peer-to-Peer Systems and Applications: State-of-the-Art-Survey and Textbook with Teaching Material*, Vol **LNCS Volume 3485**. Springer Verlag, 2005. 76, 77

[151] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli. **IP micro-mobility support using HAWAII**. Internet Draft, IETF, July 2000. (Work in progress). 40

[152] R. Ramjee, K. Varadhan, L. Salgarelli, S.R. Thuel, Shie-Yuan Wang, and T. La Porta. **HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks**. *Networking, IEEE/ACM Transactions on*, **Vol.10**(3):396–410, Jun 2002. 40

[153] B. Rankov and A. Wittneben. **On the Capacity of Relay-Assisted Wireless MIMO Channels**. In *Proceedings of Signal Processing Advances in Wireless Communications, SPAWC 2004*, July 2004. 3

[154] T. S. Rappaport. *Wireless Communications: Principles & Practice , 2nd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002. 171

[155] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. **A scalable content-addressable network**. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM. 90

[156] N. Ravi, J. Scott, L. Han, and L. Iftode. **Context-aware Battery Management for Mobile Phones**. In *PERCOM '08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 224–233, Washington, DC, USA, 2008. IEEE Computer Society. 174

[157] D. Raz, A. T. Juhola, J. Serrat-Fernandez, and A. Galis. *Fast and Efficient Context-Aware Services (Wiley Series on Communications Networking & Distributed Systems)*. John Wiley & Sons, 2006. 49, 50

[158] P. Reinbold and O. Bonaventure. **IP micro-mobility protocols**, 2003. 40

[159] T. L. Saaty. **How to make a decision: The analytic hierarchy process**. *European Journal of Operational Research*, **Vol. 48**(No. 1):9 – 26, September 1990. 61

[160] O. D. Sahin, A. Gupta, D. Agrawal, and A. El Abbadi. **Query Processing Over Peer-to-Peer Data Sharing Systems**. Technical Report UCSB/CSD-2002-28, University of California at Santa Barbara, US, 2002. 90

[161] A.K. Salkintzis. **The EAP-GPRS protocol for tight integration of WLANs and 3G cellular networks**. *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, **Vol.3**:1793–1797, Oct. 2003. 28

[162] M. Sauter. *Grundkurs Mobile Kommunikationssysteme*, chapter 4, pages 255–296. Vieweg, 2. edition, 2006. 26

[163] B.N. Schilit and M.M. Theimer. **Disseminating active map information to mobile hosts**. *Network, IEEE*, **Vol.8**(5):22–32, Sep/Oct 1994. 46

[164] J. H. Schiller and A. Voisard, editors. *Location-Based Services*. Morgan Kaufmann, 2004. 87, 88, 96

[165] A. Schmidt. *Ubiquitous Computing - Computing in Context*. PhD thesis, Computing Department, Lancaster University, November 2002. 49, 53

[166] C. Schmidt and M. Parashar. **Analyzing the Search Characteristics of Space Filling Curve-based Indexing within the Squid P2P Data Discovery System**. Technical Report TR-276, CAIP, Rutgers University, December 2004. 71

[167] H. Schulzrinne, A. G. Forte, and S. Shin. **User mobility in IEEE 802.11 networks: extended abstract**. In *MobiArch '06: Proceedings of first ACM/IEEE international workshop on Mobility in the evolving internet architecture*, pages 7–8, New York, NY, USA, 2006. ACM. 2

[168] J.M. Serrano, J. Serrat, and J. Strassner. **Ontology-Based Reasoning for Supporting Context-Aware Services on Autonomic Networks**. *Communications, 2007. ICC '07. IEEE International Conference on*, pages 2097–2102, June 2007. 55

[169] S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Prentice Hall, 2003. 88

[170] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. **Hierarchical Mobile IPv6 Mobility Management (HMIPv6)**. Rfc 4140, IETF, August 2005. 39, 40

[171] M. Stemm and R. H. Katz. **Vertical handoffs in wireless overlay networks**. *Mobile Networks and Applications*, **Vol.3**(4):335–350, January 1998. 27

[172] A. Stenzer. **Bewertung von Overlays für die Mobilitäts- und Standortverwaltung (Evaluating Overlays for Mobility and Location Management)**, 2008. Diploma thesis under the supervision of Amine Houyou, University of Passau, Februar 2008. 115

[173] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. **Internet Indirection Infrastructure**. In *SIGCOMM 2002: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 73–86, New York, USA, August 2002. ACM Press. 43

[174] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. **Internet indirection infrastructure**. *IEEE/ACM Transactions on Networking*, **Vol.12**(2):205–218, April 2004. 43

[175] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. **Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications**. In *In Proc. 2001 ACM SIGCOMM Conference*, pages 149–160, August 27-31 2001. `http://www.acm.org/sigcomm/sigcomm2001/p12-stoica.pdf`. 78, 129

[176] I. Stoica and H. Zhang. **Providing guaranteed services without per flow management**. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 81–94, New York, NY, USA, 1999. ACM. 50

[177] T. Strang and C. Linnhoff-popien. **A Context Modeling Survey**. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004. 51

[178] Y. Tang. **A Short Survey on P2P Data Indexing**. Technical report, Fudan University, Shanghai, China, 2008. 91, 93

[179] S. Tsao and C. Lin. **VGSN: a gateway approach to interconnect UMTS/WLAN networks**. *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, **Vol.1**:275–279 vol.1, Sept. 2002. 28

[180] K. Ushiki and M. Fukazawa. **A new handover method for next generation mobile communication systems**. *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, **Vol.5**:2560–2565 vol.5, 1998. 28

[181] A. G. Valkó. **Cellular IP: a new approach to Internet host mobility**. *SIGCOMM Comput. Commun. Rev.*, **Vol.29**(1):50–65, 1999. 40

[182] A. Varga. **Parameterized Topologies for Simulation Programs**. In *In Proceedings of the Western Multiconference on Simulation (WMC'98), Communication Networks and Distributed Systems (CNDS '98)*, 1998. San Diego, CA, January 11–14. 151

[183] A. Varga. **The OMNeT++ discrete event simulation system**. In *In Proc. of theEuropean Simulation Multiconference (ESM'2001)*, June 6-9 2001. 158

[184] A. Varga and G. Pongor. **Flexible Topology Description Language for Simulation Programs**. *Proceedings of the 9th European Simulation Symposium (ESS'97) at University of Passau*, **Vol.9**:225–229, 1997. 151

[185] J. O. Vatn. **An experimental study of IEEE 802.11b handover performance and its effect on voice traffic**, July 2003. 139, 154

[186] H. Velayos and G. Karlsson. **Techniques to Reduce IEEE 802.11b MAC Layer Handover Time**, June 2003. 2

[187] P. Vidales, R. Chakravorty, and C. Policroniades. **PROTON: a policy-based solution for future 4G devices**. *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on*, pages 219–222, June 2004. 60

[188] H.J. Wang, R.H. Katz, and J. Giese. **Policy-enabled handoffs across heterogeneous wireless networks**. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 51–60, February 1999. 27, 60

[189] Q. Wei, K. Farkas, C. Prehofer, P. Mendes, and B. Plattner. **Context-aware handover using active network technology**. *Comput. Netw.*, **Vol.50**(15):2855–2872, 2006. 56, 58

[190] D. Wisely, P. Eardley, and L. Burness. *IP for 3G: Networking Technologies for Mobile Communications*. John Wiley & Sons, Ltd, Baffins Lane, Chichester, West Sussex, PO 19 1UD, England, 2002. 8, 18, 25, 33, 34

[191] S. Woon, E. Wu, and A. Sekercioglu. **A Simulation Model of IEEE 802.11b for Performance Analysis of Wireless LAN Protocols**. *Department of Electrical and Computer Systems Engineering*, 2004. 153

[192] E. Wu, J. Lai, and A. Sekercioglu. **An Accurate Simulation Model for Mobile IPv6 Protocol**. *Department of Electrical and Computer Systems Engineering*, 2004. 153

[193] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma. **Performance of reliable transport protocol over IEEE 802.11 wireless LAN: analysis and enhancement**. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, **Vol.2**:599–607 vol.2, 2002. 167

[194] J. Xie. **User independent paging scheme for mobile IP**. *Wirel. Netw.*, **Vol.12**(2):145–158, 2006. 40

[195] Z. Yang and X. Wang. **Joint Mobility Tracking and Hard Handoff in Cellular Networks via Sequential Monte Carlo Filtering**. In *In Proceedings*

of the Conference on Computer Communications (IEEE Infocom, pages 968–975, 2002. 19, 41

[196] W. Ying, Z. Yun, Y. Jun, and Z. Ping. **An Enhanced Media Independent Handover Framework for Heterogeneous Networks**. *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2306–2310, May 2008. 32

[197] Z. R. Zaidi and B. L. Mark. **Real-Time Mobility Tracking Algorithms for Cellular Networks Based on Kalman Filtering**. *IEEE Transactions on Mobile Computing*, **Vol. 4**(No. 2):pages 195 – 208, March/April 2005. 41, 62, 146, 171, 172

[198] F. A. Zdarsky, I. Martinovic, and J. B. Schmitt. *Self-Organizing Systems*, Vol **4124** of *LNCS*, chapter The Case for Virtualized Wireless Access Networks, pages 90–104. Springer, Berlin Heidelberg, September 2006. First International Workshop, IWSOS 2006, and Third International Workshop on New Trends in Network Architectures and Services, EuroNGI 2006. 117

[199] Xiaowei Zhang, Javier Gomez Castellanos, and Andrew T. Campbell. **P-MIP: paging in mobile IP**. In *WOWMOM '01: Proceedings of the 4th ACM international workshop on Wireless mobile multimedia*, pages 44–54, New York, NY, USA, 2001. ACM. 40

[200] F. Zhu and J. McNair. **Multiservice vertical handoff decision algorithms**. *EURASIP J. Wirel. Commun. Netw.*, **2006**(2):52–52, 2006. 59, 60

[201] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker. **Host mobility using an internet indirection infrastructure**. *Wireless Networks*, **Vol.11**(6):741–756, 2005. 43, 44, 79, 131

Declaration

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Doktorarbeit selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle wörtlich oder sinngemäß übernommenen Ausführungen wurden als solche gekennzeichnet. Weiterhin erkläre ich, dass ich diese Arbeit in gleicher oder ähnliche Form nicht bereits einer anderen Prüfungsbehörde vorgelegt habe.

München, den 30. Juni 2009

.....

(Amine Mohamed Houyou)