Dynamic Location Management in Heterogeneous Cellular Networks

James Cowling

A thesis submitted in partial
fulfillment of the requirements for the degree of
Bachelor of Computer Science and Technology (Advanced) (Honours)

Advanced Networks Research Group
School of Information Technologies
University of Sydney
Australia

Supervisor: Dr Björn Landfeldt

October 2004

Abstract

Continued expansion of cellular networks, coupled with an increasingly restricted mobile spectrum, has established the reduction of communication overhead as a highly important issue. Much of this traffic is used in determining the precise location of individual users when relaying calls, with the field of location management aiming to reduce this overhead through prediction of user location.

Previous approaches to location management have suffered from two major weaknesses - the imposition of unrealistic assumptions on user movement and the requirement for a uniform homogeneous network; precluding implementation in real-world systems. This thesis presents a complete and novel location management scheme, addressing weaknesses present in previous proposals, while maintaining a high level of implementation feasibility.

Here the network is modelled as an abstract weighted graph, recording aggregate movement probabilities and adjusting dynamically to temporal network characteristics. Users are characterised via call arrival rate and movement factor, with these utilised by a cost function evaluating overhead for a given location area. A state machine interpretation of the graph topology is used to estimate residence time within the area, allowing the shape and size of each location area to be optimised per cell and per user.

The performance of this scheme is found to improve significantly on current systems where network and user heterogeneity is present, degrading to an optimal static location management scheme for a perfectly homogeneous network.

Acknowledgements

Many people are owed a debt of thanks for their kind assistance and support throughout my honours year. From an academic perspective I would like to acknowledge Natalie Kolodziej for her research efforts into dynamic location management, providing an inspiration for the work presented here; Harry for the invaluable assistance in deriving a matrix formulation of location area residence time; Adam from Optus for his kind loan of an N-Gage to allow me to conduct my survey into modern GSM networks; and Justin Taheri for providing his MobileManager trace generator to assist in my analysis, kindly modifying his program at late notice to suit my needs.

Many others have provided support on a less tangible but equally important level. Michael Till whose support and guidance outweighed his proficiency for distraction many times over. Nick for regularly venturing to the labs to provide humanitarian aid. Friends and fellow students of LG45 - Matt, Daniel, Chris, Rhys, Megan, Dave, Tara, Harry, Paul, Lang and Tore, for making this year most unexpectedly enjoyable. Fabs for moral support and entertainment at 4am each morning, an endless supply of 2 minute noodles, and for being someone whom I spent more time sitting next to at a desk than I did eating and sleeping combined. Hugh for encouraging me to take time off from study, having the same opinion as me on every issue in the world, and teaching me how to play Karma Police on guitar.

I would like to thank Trent for not only being a great friend, but also for providing an inspiration to extend myself in all my academic work, culminating in this thesis.

My family for tolerating the erratic schedule of an honours student, especially my mother for her valiant proof-reading efforts and continued support and concern.

Finally I would like to thank Bjorn for being tremendously supportive and inspirational, both on an academic and personal level; for encouraging me to continue with graduate study without ever having to say it.

Definitions

The following select terms are used throughout this document and replicated here for reference.

- **Base Station**

  A tower or antenna transmitting and receiving radio signals over a cell in a wireless network.

- **Base Station Controller (BSC)**

  An agent performing functions on behalf of a group of base stations. The BSC handles the allocation of radio channels, controls handovers, performs paging and interfaces with the central network and HLR.

- **Cell**

  A geographical area serviced by a base station in a wireless network, also used to refer to one or more collocated base stations. Cells are the 'building blocks' of a cellular network, with overlapping cells defining the coverage area of a particular network.

- **Global System for Mobile Communication (GSM)**

  The dominant standard for second generation mobile phone communication, defining the protocols for communication between mobile devices and network cells.

- **Handoff**

  The process of transferring an in-progress call from one cell or base station to a neighbouring cell without interruption.

- **Home Location Register (HLR)**
  The central database in a cellular network, containing information on all subscribers to a particular carrier. This database also contains a record of each user's location, used to route calls to the correct cell.

- **Location Area (LA)**
  A group of neighbouring cells combined to form a larger *meta-cell*. Devices are free to move within this Location Area without performing a Location Update. Location Areas may be fixed, as in current static schemes, or allocated dynamically on a Location Update.

- **Location Management (LM)**
  The maintenance of a record of cell locations for devices in a mobile network. The study of Location Management aims to reduce the net cost involved in maintaining this information.

- **Location Update (LU)**
  Performed by a device in a wireless network to inform the network of the cell in which it resides. This Location Update is usually performed only when leaving the Location Area previously assigned to the device.

- **Paging**
  Under a Location Area scheme, the network does not know the precise location of a device, only its general area. Paging is performed on an incoming call and involves sending a message to all cells in the Location Area to determine which one contains the destination device.

- **Spectrum**
  A portion of the electromagnetic spectrum containing a limited frequency range within which a mobile device may communicate. It is vital that multiple signals transmitted on the same frequency do not interfere and hence the allocation of sections of this spectrum

is governed by regulatory bodies. A communications provider must purchase a licence for a particular frequency band within this spectrum to broadcast cellular data.

- **Subscriber Identity Module (SIM)**

  A small *smart card* used in mobile phones operating under the GSM standard. This SIM card contains user identification information, as well providing storage space for phone numbers and associated data.

- **Third Generation (3G)**

  A new wireless communication specification replacing second generation technologies such as GSM. Third generation cellular networks provide for high-speed data access in addition to audio communication, with goals of high-quality multimedia and advanced global roaming.

Contents

List of Figures

Chapter 1

Introduction

This thesis presents a *per-user dynamic location management scheme*, for use within arbitrary *heterogeneous cellular networks*. The motivation and goals guiding the development of this system are discussed in this chapter. The specific research contributions of this project are outlined, along with an overview of the structure of the remainder of this document.

1.1 Location Management

A cellular communication system must track the location of its users in order to forward calls to the relevant cell within a network. A naïve solution to this problem would involve users informing the network of their new location as they transition between cells. Unfortunately such an approach is infeasible, with the subsequent update communication quickly overwhelming a network.

Instead cells within a network are grouped into *Location Areas* (LAs). Users are free to move with a given location area without updating their location, informing the network only when transitioning to a new LA. If a call is to be forwarded to a user, the network must now page every cell within the location area to determine their precise location. Network cost is incurred on both location updates and paging, the balance of these defining the field of *Location Management* (LM), discussed in more detail in the following chapter.

The frequency of updates and paging messages relates closely to user movement and call arrival rates, along with network characteristics such as cell size. Dynamic LM schemes aim to assign location areas to users based on these parameters, achieving an optimal balance between update and paging cost, minimising total overhead in the network. Per-user dynamic schemes consider individual user characteristics, while aggregate dynamic schemes take into account only average user parameters at each location, providing a lesser degree of granularity. Both schemes contrast strongly with currently implemented static schemes, where the grouping of cells into location areas is pre-determined, irrespective of user and network characteristics.

1.2 Motivation

Wireless spectrum is an expensive commodity. Table 1.1 shows the total revenue earned by European governments in auctioning wireless spectrum to communications providers for the 3G mobile communication system. The total cost of over 100 billion Euros represents an enormous financial undertaking - network operators in the United Kingdom alone spent the equivalent of over 1000 Australian dollars for ever man, woman and child in the entire country. As sections of usable spectrum are committed to network carriers, prices may be expected to increase further, presenting one of the primary business concerns for communication providers. It should be noted that this cost is merely in securing the *rights* to communicate on a given frequency range, the equipment to do so requires significant additional financial outlay.

With wireless spectrum such an extremely expensive resource, it is of prime importance to minimise unnecessary network bandwidth use, in doing so minimising the spectrum requirements for a network operator. With the management of user location absorbing significant proportions of this spectrum, great cost reductions are available through efficient location management implementation. The continued growth of wireless communication systems, and expansion of network subscription rates, signals increased demand for this efficient location management.

Currently implemented location management schemes are purely static - all users in a given region are assigned the same location area regardless of their characteristics. Such an

Country	Cost per Capita (Euros)	Total Cost (million Euros)
Austria	100	818
Belgium	45	466
Denmark	95	514
Germany	615	50,691
Greece	45	479
Italy	240	13,934
Netherlands	170	2,697
Switzerland	20	149
United Kingdom	650	39,176
	Total:	**108,924**

Table 1.1: Approximate Revenues from European 3G Mobile Spectrum Auctions 2000-2001 [Kle01]

implementation is significantly suboptimal. Dynamic location management schemes aim to reduce the cost of static schemes by assigning appropriately sized and shaped location areas depending on individual user or network characteristics.

Current dynamic location management schemes, while reducing total location management cost, have significant weaknesses. The great majority of effective schemes sacrifice a large degree of feasibility by aiming to use complex heuristics to predict user location, precluding their implementation in real communication networks. Efficient LM schemes avoid the use of complex prediction schemes, but consider only aggregate user characteristics, ignoring a wealth of user data which may be used in reducing location management cost.

Location management proposals in current research also make a number of unrealistic assumptions, the details of which will be considered in the following chapter, primarily:

- The assumption of random or simplistic individual user movement
- The assumption of a uniform, infinitely-large, hexagonal grid network topology

The unrealistic assumptions made in current dynamic location management, along with a lack of focus on feasible implementation, have prevented their implementation in developing cellular network standards. A per-user dynamic LM scheme, able to be implemented efficiently in arbitrary networks, would stand to benefit network carriers tremendously.

1.3 Project Goals

The primary aim of this project is simply to reduce the cost incurred on cellular network operators in managing the location of users. All developments are proposed in the context of this aim - location management techniques are only of practical significance if the cost of their implementation is shadowed by the cost reductions provided by the schemes themselves. Underscoring this philosophy, the research presented will take a departure from complex heuristic-based location management techniques, with an emphasis on *highly efficient* dynamic location management capable of *feasible implementation* in *real cellular networks*.

A per-user dynamic location management scheme is presented, adhering to this major objective. The aim of feasible cost reduction may be realised through the following goals:

- Develop a dynamic location management scheme, capable of application to arbitrary heterogeneous networks
- Implement an efficient machine learning system to automatically optimise system parameters, capturing temporal network characteristics
- Formulate cost metrics in terms of simple user and network parameterisation, allowing feasible implementation in current networks
- Avoid assumptions on network topology and user behaviour, permitting applicability to a wide range of future networks

Aside from these goals, the proposed scheme is claimed to achieve specific cost-reduction performance greater than currently available dynamic schemes, owing to a focus on per-user network behaviour and a rejection of unrealistic network assumptions.

1.4 Contributions

Many individual developments and contributions are required in the satisfaction of the aforementioned project goals. These contributions, in the context of this project, are summarised as follows:

- Identification of a number of weaknesses in previous location management systems, to be addressed by the proposed location management scheme.
- An architecture for representing a cellular network within a location management system, comprised of cells containing dwell time and weighted edges reflecting the probability of movement between neighbouring cells. This architecture is completely dynamic and self-generating, able to accurately reflect temporal network characteristics and network changes using simplified machine learning with minimal computational requirements.
- A search-based algorithm for determining the optimal shape of a location area, based on aggregate movement probabilities and mean residence time at each cell. This system is able to implicitly capture concepts of cell sizes, network boundaries and directed regions, without manual definition of such characteristics.
- A system for characterising user movement independently of 'ping-ponging' effects prevalent in modern networks, avoiding the collapse of previous systems under such conditions.
- Formulation of a generalised cost function evaluating the relative overhead expected for a given location area and user characteristics. This cost function may apply to any per-user dynamic location management scheme where user residence time within a location area may be estimated.
- A methodology for estimating the time a user will reside within a given location area, performed via sparse matrix computation. The graph abstraction of each location area is considered as a state machine with execution times defined by dwell time at each cell, with the time taken to exit the state machine defining the average time spent within the area by a user.
- Evaluation of the efficiency of the proposed location management scheme and feasibility for implementation in current networks.

- Implementation of a software tool to record network cell and location area information in the form of trace data from mobile phones running the Symbian operating system.
- A survey of modern GSM networks, analysing their parameters and influence on user behaviour as compared to traditional metrics.
- An evaluation of currently available location management benchmarks, critiquing their applicability to modern cellular networks.
- Implementation of a comprehensive cellular network simulation package, operating a number of candidate location management schemes and able to accept trace data from a wide range of sources. This package is also capable of automatically generating trace input for a variety of user characteristics and network topologies.
- Computation and analysis of results evaluating the performance of the proposed location management scheme under a wide variety of network conditions.
- The proposal and evaluation of a number of methods for efficiently transmitting a dynamic location area to a device.

1.5 Thesis Structure

This thesis covers the development, testing and analysis of a dynamic location management scheme, capable of implementation in arbitrary networks and optimised per-user. The following chapters will build a firmer context for the development of this scheme, detail the location management proposal, and evaluate its performance under a variety of scenarios.

Chapter 2 presents a more thorough overview of previous research into location management, isolating a number of key weaknesses to be addressed in the development of the proposed scheme. It concludes by formulating these weaknesses into a motivated research direction, signalling the requirements for the design of the location management system.

Chapter 3 comprises the major contribution of this thesis - the design and justification of a complete per-user dynamic location management system for implementation in heterogeneous cellular networks. The development of this scheme is presented incrementally, from an abstract weighted graph network representation to the final estimation of user residence time within a given location area. Implementation feasibility is considered throughout this development, to ensure a scheme capable of realistic implementation in a modern cellular network.

Chapter 4 details a survey of current GSM communication networks, with an evaluation of possible benchmarks used to test the performance of the proposed location management scheme. The results of trace data are analysed and compared with previously held beliefs about cellular network characteristics, to re-establish an accurate set of requirements for a realistic location management implementation.

Chapter 5 examines the performance of the proposed per-user dynamic location management system, in comparison with less complex aggregate dynamic and traditional static schemes. These comparisons are conducted over a range of performance benchmarks, including both modern and traditional artificial traces, along with real GSM network traces.

Chapter 6 investigates the specific performance impact of communicating complex location areas to users as required by a dynamic location management scheme. Various terse location area representations are proposed to minimise this additional overhead, with their spatial scalability and feasibility examined.

Chapter 7 signals the direction of future work into the design of an optimal per-user dynamic location management scheme. A number of concrete goals are presented here, with the intention of addressing each during a continued period of research in the near future.

Chapter 8 concludes the location management proposal by summarising its contributions and affirming its significance in the context of modern cellular networks.

A number of appendices follow, providing supporting information to the research and development presented here, as referenced within the text. The presentation of the proposed dynamic location management system will begin with a discussion of current developments in location management.

Chapter 2

Background

As mobile devices move between cells in a network they must register their new location to allow the correct forwarding of data. Continual location updates can be a very expensive operation, particularly for users with comparatively low call arrival rates. This update overhead not only puts load on the core (wired) network but also reduces available bandwidth in the mobile spectrum. Importantly, unnecessary location updating incurs heavy costs in power consumption for the mobile device.

The study of location management aims to reduce the overhead required in locating mobile devices in a cellular network. The two sources of network cost involved in location management are *location updating*, informing the network of a devices location, and *paging*, polling a groups of cells to determine the precise location of a device [BNIM95]. Careful application of location update and paging strategies must be made to ensure minimal communication and processing cost. An additional area of consideration in location management is the mobility model used to estimate user movement in the network, aiding in the optimisation of location updating and paging schemes.

With mobile user subscription increasing rapidly and a continually restricted mobile spectrum available [AMH+99], the problem of effective location management has become one of prime significance. The main approaches to location updating: paging and mobility modelling, will be discussed here. These provide the impetus for a complete and effective approach to location management.

2.1 Location Update

A location update is used to inform the network of a mobile device's location. This requires the device to register its new location with the current base station, to allow the forwarding of incoming calls. Each location update is a costly exercise, involving the use of cellular network bandwidth and core network communication; including the modification of location databases [AMH+99]. A wide variety of schemes have hence been proposed to reduce the number of location update messages required by a device in a cellular network.

Location update schemes are often partitioned into the categories of static and dynamic [BNIM95, AMH+99, WL00, ADM98]. Static schemes offer a lower level of cost reduction but reduced computational complexity. Dynamic schemes adjust location update frequency per user and are hence able to achieve better results, while requiring a higher degree of computational overhead.

2.1.1 Static Location Update Schemes

Static schemes define the frequency and occurrence of location updates independently from any user characteristics. Such static mechanisms allow efficient implementation and low computational requirements due to the lack of independent user tracking and parameterisation.

2.1.1.1 Always-update vs. Never-update

The always-update strategy is the simplest location update scheme; performing a location update whenever the user moves into a new cell. The network always has complete knowledge of the user's location and requires no paging to locate the user when an incoming call arrives.

The always-update scheme performs well for users with a low mobility rate or high call arrival rate. It performs quite poorly for users with high mobility however, requiring many location updates and excessive use of resources. While not used in practice, this scheme forms the basis of many more complex location management mechanisms, such as location area [OOYM91] and profile-based update [Lin97].

The never-update scheme is the logical counterpart to always-update, never requiring the mobile device to update its location with the network. This entails no location update overhead but may result in excessive paging for a large network or for high call arrival rates. These two strategies represent the extremes of location management - always-update minimising paging cost with never-update minimising location update cost [SZ03]. The two extremes are often combined to form a more comprehensive location management strategy, catering for differences in user and network characteristics [AMH+99, WL00].

2.1.1.2 Reporting Cells

[BNK93] proposes a location update scheme where the mobile device updates its location only when visiting one of a set of predefined *reporting cells*. This scheme does not depend on the individual movement characteristics of the user, but simply the arrangement of these reporting cells in the network. When an incoming call arrives for the mobile device, a search must be conducted around the vicinity of the last reporting cell from which the user updated their location [AMH+99, WL00].

The reporting cell topology may be bounded or unbounded, as displayed in Figures 2.1 and 2.2. The unbounded approach requires a smaller number of reporting cells, in turn reducing the number of redundant location updates. This however requires an intelligent paging scheme to handle the unbounded search space. Such schemes require additional overhead and are discussed further in Section 2.2.3.

The performance gains achievable with a reporting cell topology are somewhat limited. Without direct consideration of the movements of users, it is not possible to assign reporting cells in an optimum arrangement. Even with knowledge of the network, it is shown in [BNK93]

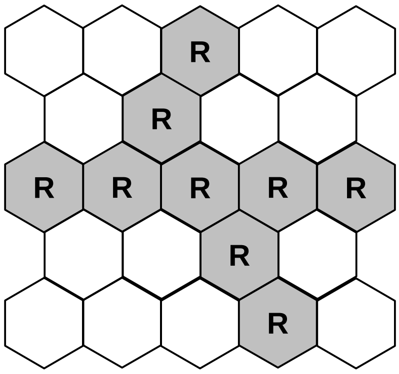


Figure 2.1: Bounded reporting cell configuration

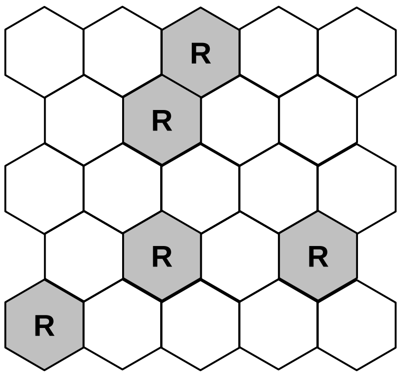


Figure 2.2: Unbounded reporting cell configuration

that the selection of an optimum set of reporting centres is an NP-complete problem. Two extremes of poor performance are possible, owing to the unpredictability of user movement. When a mobile device continually passes over a reporting cell, excessive location update messages are generated, introducing a high level of overhead. Conversely, if a mobile device does not roam into any of the reporting cells, its location will never be updated and hence incur a high paging load [SZ03].

[SZ03] proposes a dynamic reporting cell alternative to address the weaknesses in the static scheme. Here a set of parallel reporting cell topologies is used, to vary the configuration per-user. These topologies then evolve through the use of genetic algorithms, to optimise the location of reporting cells for current mobility patterns. While such a scheme allows a near-optimal arrangement of reporting cells to be derived, the additional computational overhead involved detracts from the efficiency and ease of implementation that lends appeal to the standard reporting cell approach.

2.1.1.3 Location Areas

The location area topology is widely used to control the frequency of location updates, both in the current GSM [Rah93] and IS-41 [EIA91] telecommunications architectures. Here the network is partitioned via the amalgamation of groups of cells into larger meta-cells, or *location areas*. The scheme then functions very similarly to the always-update mechanism - mobile devices only update their location when leaving their current location area [OOYM91]. This partitioning is shown in Figure 2.3, with four separate location areas.

If the network knows the users current location area, the paging required to locate a user is confined to the meta-cell within which the user resides. The location update process may be instigated periodically or, more commonly, on location area boundary crossings.

The periodic location updating scheme is the simplest to implement, merely requiring the mobile device to send a location update message containing its current location area at regular time intervals [Tab97]. The methodology used here captures none of the details of user movement however, and enforces an update frequency that may be highly unsuitable given

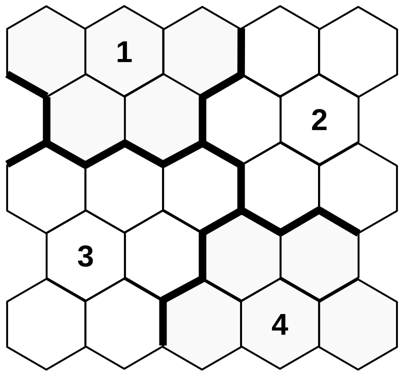
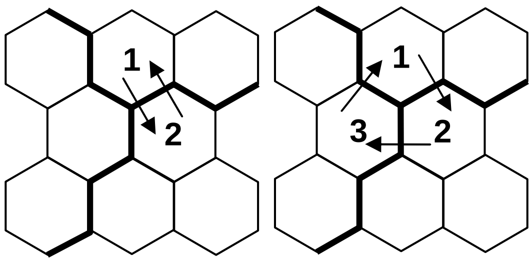


Figure 2.3: Network partitioned into location areas

the users current rate of movement. It also offers no guarantees that the network will have knowledge of the exact location area the user resides in, requiring sophisticated paging across multiple location areas. The boundary crossing method is more precise; updating the user location only when crossing to another location area. This method has its own inherent weaknesses however, particularly when a user makes multiple crossings across a location area boundary [EGO02].



(a) 2-cell configuration

(b) Generalised configuration

Figure 2.4: Cell ping-pong effect

The ping-ponging effect, illustrated in Figures 2.4(a) and 2.4(b), is the major weakness of location area schemes. Here a user moves repeatedly between the boundaries of two or more location areas, inducing a high location update rate with comparatively low physical mobility. A number of schemes have been proposed to address this problem, such as the Two Location

Area (TLA) [Lin97] and Three Location Area (TrLA) [EGO02] mechanisms. These assign multiple location areas to a mobile device, requiring location update only when one of the respective two or three location areas has been exited. These schemes are shown to provide increasing levels of performance, although incur additional computational overhead.

2.1.2 Dynamic Location Update Schemes

Dynamic location update schemes allow per-user parameterisation of the location update frequency. These account for the dynamic behaviour of users and may result in lower location management costs than static schemes. Unlike static location management strategies, a location update may be performed from any cell in the network, taking into consideration the call arrival and mobility patterns of the user. Dynamic schemes have been the area of much active research, the contributions of which are summarised following.

2.1.2.1 Threshold-based

In threshold-based schemes each mobile device maintains a particular parameter, updating its location when the parameter increases beyond a certain threshold. The update threshold may be optimised on a per-user basis, dependent on a user's call-arrival and mobility rate [AHL96]. The most common thresholding schemes are time-based, movement-based and distance-based; evaluated in [BNIM95] and [AMH+99].

Time-based Update The time-based strategy requires that users update their location at constant time intervals. This time interval may then be optimised per-user, to minimise the number of redundant update messages sent. This only requires the mobile device to maintain a simple timer, allowing efficient implementation and low computational overhead.

Figure 2.5 illustrates this scheme, with the updates (U1, U2 and U3) performed at each time interval Δt, regardless of individual movements. It was found in [NL98] that signalling load may be reduced using a time-based scheme, with the network additionally able to determine if a mobile device is detached if it does not receive an update at the expected time. [Ros96]

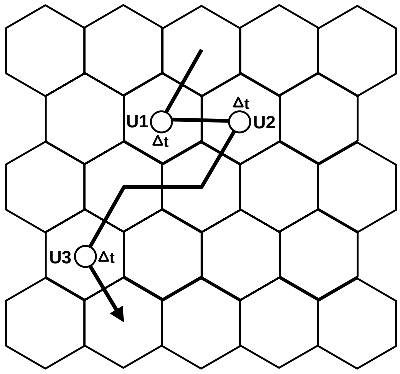


Figure 2.5: Time-based location update

analyses the performance of the time-based location update scheme, independent of user mobility constraints, and finds it to outperform the location area method used in current systems. The time-based scheme does however entail a high degree of overhead in a number of situations, such as when a user has only moved a very small distance or has not moved at all [SZ03].

Movement-based Update The movement-based update scheme requires mobile devices to update their location after a given number of boundary-crossings to other cells in the network. This boundary-crossing threshold may be assigned per-user, optimised for individual movement and call arrival rates [AHL96].

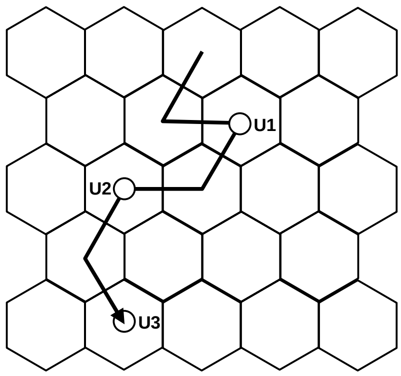


Figure 2.6: Movement-based location update

Figure 2.6 shows a movement-based scheme, with a movement threshold of two. Here the device updates its location every two crossings between cells. The required paging area is restricted to a neighbourhood of radius equal to the distance threshold around the last updated location. The paging area requirement is reduced through this scheme, although unnecessary updates may still be performed as a result of repeated crossings over the same cell boundary. [BNIM95] finds the movement-based scheme to perform better than the time-based scheme under a memory-less (random) movement model. Under Markovian analysis, the time-based scheme is found to perform better for special cases of extremely low update rates, with the movement-based scheme performing slightly better for the majority of user classes.

Distance-based Update In a distance-based scheme the mobile device performs a location update when it has moved a certain distance from the cell where it last updated its location. Again, this distance threshold may be optimised per-user according to movement and call-arrival rates.

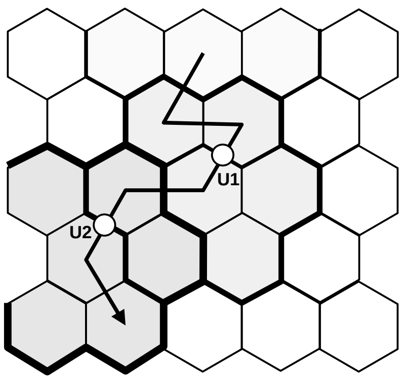


Figure 2.7: Distance-based location update

A distance-based update system is shown in Figure 2.7. Here the location is only updated when the user travels beyond a certain radii from the previous update location. This scheme has the benefit of not requiring an update when a user repeatedly moves between a small subset of cells, provided these cells reside within the distance-threshold radius. [BNIM95] and [AMH+99] find the distance-based scheme to significantly outperform the time- and movement-based schemes under both simulation and theoretical analysis.

Distance-based update strategies are quite difficult to implement in a real-world network [FI03]. Each mobile node is required to track its location and determine the distance from the previously updated cell. This not only requires the device to retain information about its starting position, but also to possess some concept of a coordinate system, allowing the calculation of distance between the two points. This coordinate system is a non-trivial requirement in a heterogeneous network, where cell adjacencies and distances may not be clearly defined.

2.1.2.2 Profile-based

Under a profile-based scheme the network maintains a profile for each user in the network, based on previous movements, containing a list of the most probable cells for the user to reside within [Lin97]. On a location update the network sends this list to the mobile device, forming what may be considered a complex location area. The mobile device updates its location only when entering a cell not contained in the list. [XTG93] found that when movement exhibits a medium to high predictability, the location management cost is lower than that provided by schemes with a more general representation of location area. When low predictability is encountered, the high overhead of sending a large cell list to users outweighs the cost reduction provided by the profile-based scheme.

2.1.2.3 Adaptive

The group of adaptive location update schemes comprises a large number of current developments in location management. Adaptive schemes take into consideration a variety of user parameters when assigning a location area to a mobile device. [AMH+99] describes an ideal location management scheme as one which adjusts on a per-user basis, as is provided by adaptive schemes.

The predictive distance-based update scheme [LH03], a variant of adaptive location updating, predicts a mobile's future location based on the location and velocity information registered during an update. The shape of the assigned location area reflects the mobility patterns of the user, while the size of the area varies as a function of the incoming call rate.

An activity-based location update scheme is proposed in [SK99], based on the same framework as predictive distance-based update. The frequency of each cell movement between adjacent cells is measured along with the residence time at each cell. From this information the likelihood of residence in each cell is evaluated, with each cell added in decreasing probability order until the maximum location area size is reached. The size of this location area is derived from the movement rate and call arrival history of each node.

[Ros99] examines the performance of various location update schemes by examining the reduction in variability of paging and registration cost under varying mobile characteristics. The use of adaptive location management schemes is found to consistently reduce location update overhead. The main disadvantage of adaptive updates can hence be seen as the excessive implementation complexity and computational overhead required by both the network and individual mobile nodes.

2.2 Paging

While mobile devices perform updates according to their location update scheme, the network needs to be able to precisely determine the current cell location of a user to be able to route an incoming call. This requires the network to send a paging query to all cells where the mobile device may be located, to inform it of the incoming transmission. It is desirable to minimise the size of this paging area, to reduce the cost incurred on the network with each successive paging message [FI03]. Ideally the paging area will be restricted to a known group of cells, such as with the currently implemented location area scheme [OOYM91]. An optimum paging area size calculation involves a trade-off between location update cost and paging cost. This technique is used in many location management schemes to reduce the location management costs incurred.

The most commonly used paging schemes are summarised below. These have seen extensive use in real-world telecommunications networks [FI03].

2.2.1 Simultaneous Paging

The simultaneous paging scheme, also known as blanket paging, is the mechanism used in current GSM network implementations [WL00]. Here all cells in the users location area are paged simultaneously, to determine the location of the mobile device. This requires no additional knowledge of user location but may generate excessive amounts of paging traffic [Lee95].

Implementations of simultaneous paging favour networks with large cells and low user population and call rates. This scheme does not scale well to large networks with high numbers of users, necessitating the development of more advanced paging techniques.

2.2.2 Sequential Paging

Sequential paging avoids paging every cell within a location area by segmenting it into a number of *paging areas*, to be polled one-by-one. It is found in [RY95] that the optimal paging mechanism, in terms of network utilisation, is a sequential poll of every cell in the location area individually, in decreasing probability of user residence. The individual delays incurred in this scheme may be unacceptable however, and hence it is suggested that paging areas are formed from a larger number of cells. The number of cells per paging area is a factor which needs to be optimised and may lead to excessive call delays, particularly in large networks.

The order by which each area is paged is central to the performance of the sequential paging scheme. [RY95] suggests several methods to determine the ordering of paging areas in a sequential scheme. The simplest ordering constraint is a purely random assignment, where each paging area is polled in a random order. While this reduces the total number of polling messages over a blanket scheme, it is far from optimal. Schemes favouring paging areas located geographically closer to the previously updated location are found to further reduce the total number of paging messages required. These schemes necessitate knowledge of the geographical structure of the network however, and may perform poorly for high movement rates.

2.2.3 Intelligent Paging

The intelligent paging scheme is a variation of sequential paging, where the paging order is calculated probabilistically based on pre-established probability metrics [RY95]. Intelligent paging aims to poll the correct paging area on the first pass, with a high probability of success. This efficient ordering of paging areas requires a comprehensive knowledge of user residence probabilities.

[BNIM95] discusses that the success of an intelligent paging scheme hinges on the ability of an algorithm to calculate the probability of a user residing in each cell of the current location area. An algorithm to calculate the paging area ordering based on a probability transition matrix is presented in [GR98], claiming to result in the optimal poll ordering. The computational overhead involved in this scheme is quite high however, requiring the computation of an $n \times n$ matrix for a system with n cells, and hence is infeasible for a large cellular network.

2.3 Mobility Models

A sophisticated location management scheme, such as those used in adaptive location areas, must rely on a mobility model to simulate the movements of users through a cellular network [SK99]. Such mobility models define the rate and direction of user movement through a system, in effect dictating the optimum number of location update and paging messages for that scheme [KSS01]. A comprehensive implementation of a location management scheme must utilise a model of user movement capable of accurately representing real-world users of the system. A variety of commonly used mobility models is discussed below.

2.3.1 Commonly Used Models

2.3.1.1 Random walk

The random walk mobility model is regularly used to model the movements of users in a cellular network. It assumes that the direction of each user-movement is completely random, and hence each neighbouring cell may be visited with equal probability. This model is easily implemented as it requires no state information to predict the next cell occupied by a user. [BNIM95] refers to random walk as the memory-less movement model owing to this property.

[CS04] presents a two-dimensional random walk model, capable of representing both cell crossing rate and cell residence time, for either square or hexagonal cells. This model represents a large amount of movement information while remaining highly tractable, due to the random approach. While the accuracy of a random model is higher than may be expected [CLSS00], given the complexity and irregularity of real-world networks the simplistic scheme fails to capture sophisticated movement patterns exhibited by real users.

2.3.1.2 Fluid flow

The fluid flow model views the system from a macroscopic perspective, representing the aggregate movement patterns of users [BNK93]. While this model can produce accurate representations of average boundary crossings per unit area, it does not consider the movement of individual users. This system-wide approach is hence ideal for optimising total network utilization but not appropriate from a user perspective, considering only aggregate rates.

2.3.1.3 Markovian

The Markovian mobility model defines distinct probabilities for movement from a given cell to each of its neighbours [SZ01]. These probabilities are dependent on individual user movement histories. This scheme is based on the assumption that a user moving in a given direction will continue in a similar direction with greater probability than a divergence from course.

Markovian models are able to capture movement patterns well but need an established concept of user movement from which to base probabilities.

The evaluation of movement vectors is a computationally expensive operation and has limited the application of such models. [ALLC00] examines the large number of states required to maintain a comprehensive Markovian model. The poor scalability of this state space greatly restricts the use of Markovian models and hence a simplification of the state representation is proposed, exploiting symmetry in the assumed simple hexagonal topology.

2.3.1.4 Activity-based

Activity-based models are the most recent and comprehensive in representing movement throughout a cellular network. They aim to reflect individual user behaviours based on parameters such as time of day, location and destination.

[KSS01] proposes an activity-based model using input data from a recorded trip survey. Each activity selected has an associated time, duration and location. Time is based on the previous activity for a user, duration is determined from the distribution of trip durations in the survey, and the location selected based on census data storing destinations for various activities. This model is able to capture highly complex user movements and interactions, yet requires a high level of computation and large set of input data.

The random-waypoint scheme [BMJ+98] defines movements between physical locations, with dwell time at each destination selected probabilistically. This simplifies the activity-based modelling scheme to a level which may be feasibly implemented. Random-waypoint scheme has since been described as the de facto standard in mobile computing research [YLN03]. Such activity-based schemes require careful selection of parameters to produce realistic results - [YLN03] finds that a naïve implementation of the random waypoint scheme may give highly unstable results in practice.

2.3.2 Challenges to Mobility Modeling

Previous research has highlighted the weaknesses in using an arbitrary mobility model on which to base development and evaluation of a location management scheme [KSS01]. The performance of location management schemes, particularly those focusing on location updating and paging overhead, is shown to vary highly with the mobility model in use. A comparison of the performance garnered from activity-based and random-walk mobility models shows the results for a variety of location management schemes to vary significantly. It is suggested that simplistic random models be avoided in favour of the more accurate and consistent representation produced by an activity-based scheme.

Recent developments have also shown weaknesses inherent with activity-based schemes. [YLN03] examines the performance of the current random-waypoint activity-based scheme, and find the results generated by its use to be unreliable. A formal analysis of this model is used to show that it fails to provide a steady state and that average predicted speed consistently decreases over time. While improvements have been suggested to assure steady-state performance, it is clearly demonstrated that compromises must be made in efforts to predict user movement from stochastic models.

The mobility models in current use base predictions on movement patterns of users throughout physical space. While continual increases in the complexity of mobility models [KSS01] allow greater accuracy in predicting the physical movement of users, they do not directly relate to movement throughout the *network*, based on the cell currently housing a mobile device. Multiple base-stations may cover a single physical location, a complexity not captured by the models currently in use. [CLSS00] suggests that to pre-allocate resources in reality, the model should predict the actual *cell* that the user will next connect, rather than the physical location they will move to. Here a number of cell-movement traces were recorded from a highly directional, and hence predictable, physical movement source a train journey from central Sydney to its outer suburbs. It was found that the actual movement between cells varied significantly, with much lower correlation to physical movement vectors than assumed in the application of current mobility models.

2.4 Research Motivation

2.4.1 Unrealistic Assumptions

A wide variety of location management schemes have been proposed to reduce the cost of locating devices in a cellular network. While reducing the cost involved in an idealised theoretical environment, they commonly make two major assumptions [Kol03]:

- Assuming that the user's movement pattern within the network is random, and
- Assuming an infinitely large network of uniform hexagonal cells.

Such assumptions simplify the analysis and development of location management schemes significantly, yet are often not an accurate reflection of real-world cellular networks. These assumptions will be given brief attention below.

2.4.1.1 Random User Behaviour

User motion is clearly not randomly motivated, a fact that is exploited by complex activity-based movement models. Random-walk models are still commonly used in implementation however, owing to their computational tractability [CS04]. While random mobility models produce a reasonable approximation to user behaviour in extremely dense network regions, they lack the capability of describing movement in more directional network segments, such as highways and tunnels.

[CT02] proposes a directionally-biased random mobility scheme, and evaluates its performance for various user trajectories. It is found that the directional model performs well for highly directed user movements, but sub-optimally when user movement is more random. An individual user may have periods of highly directional movement in certain network conditions, such as along a highway, while seemingly random movement patterns in a more dense network such as the CBD of a city. The authors introduce a *selective-prediction* scheme, which uses the complex predictive movement scheme only in a highly directive region, reverting to a random process in other areas. Such flexibility in modelling approach is required in modern cellular networks, containing regions of varying characteristics.

2.4.1.2 Uniform Network Topology

Most research efforts into location management represent the network as an infinite hexagonal grid [AHL96], to simplify the development of tractable movement models. These hexagonal cells tessellate without gap or overlap, and approximate a circular shape, being the ideal power coverage area for a base station transmitter [Lee95]. In reality a cellular network map is far more complex, as contrasted in Figures 2.8 and 2.9.

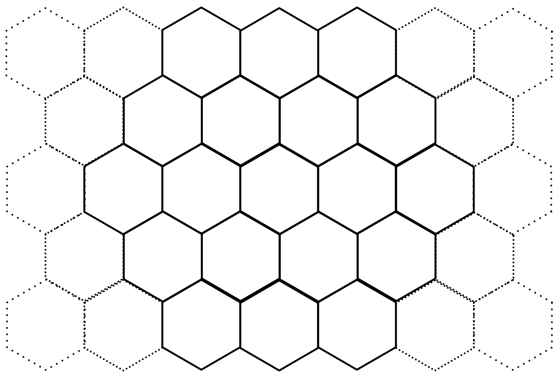


Figure 2.8: Infinite hexagonal grid topology

Realistic networks may contain complex boundaries, cells of various size and shape, and multiple overlapping cells covering a single physical area. Network boundaries may have a strong effect on user movement, biasing movement away from geographical boundaries. Schemes relying on a concrete concept of cell neighbourhood, or a simplistic coordinate system, often break down when attempts are made to apply them to a heterogeneous network where hexagonal network coordinates no longer apply.

2.4.2 Research Proposal

As an avenue for research it is desired to capture the network structure implicitly within the mobility model, in effect abstracting away the details of cell density, directionality and

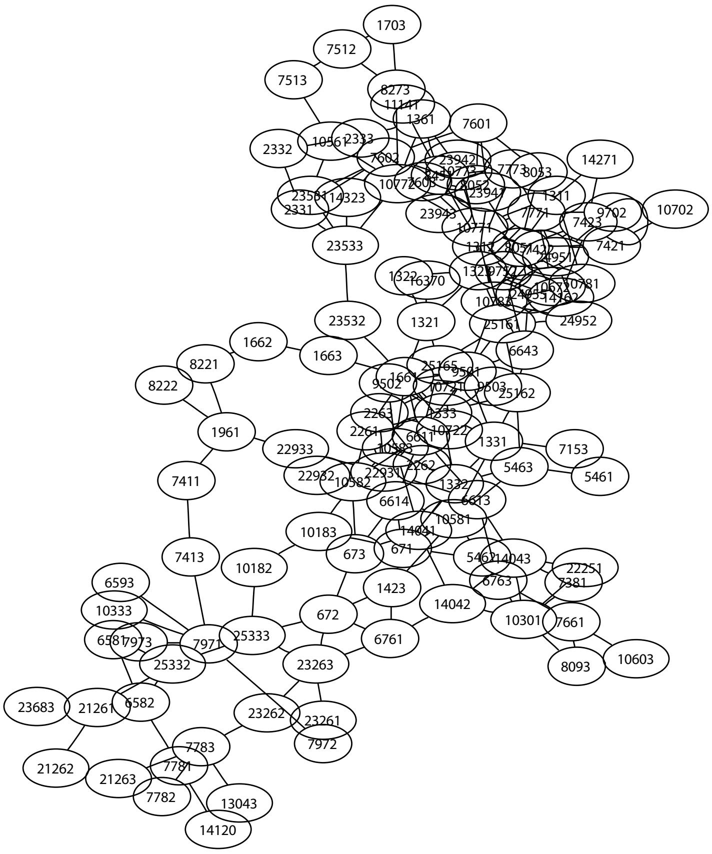


Figure 2.9: Graph model of a typical GSM network

shape. Such a procedure would result in a scheme capable of application to any arbitrary heterogeneous network, while adapting to the characteristics and boundaries of each region within the network. A base station capable of learning the mobility characteristics of its region, and the relative probabilities of movement to each neighbouring cell, would be able to dynamically record this information. Importantly, if each cell has direct knowledge of surrounding movement probabilities, this may form the mobility model used by the system optimised per region and per time of day, rather than relying on an approximate, and possibly flawed, arbitrary mobility model.

A learning network would in effect provide a Markov model with complete and accurate knowledge of movement probabilities, while entailing low computational overhead. The lack of inherent size or shape in the network, as effectively represented by a weighted graph of probabilities, abstracts the model away from physical space. This will allow implementation of the scheme in an arbitrary network, enforcing no restriction on regular cell layout or lack of network boundaries.

Such a scheme will directly address the two main short-comings of current location management mechanisms - assumed random movement and uniform network topology. The removal of these two limiting assumptions will allow feasible implementation of location management proposals in real-world networks, greatly reducing current cost overheads.

Chapter 3

Design

The development of the proposed Location Management scheme is guided by two major design goals, each addressing a traditional weakness in LM proposals as highlighted in Section 2.4.1. The removal of such weaknesses permits a scheme able to be implemented in real cellular communication networks, and one deserving of consideration for legitimate network implementation. The design goals may be expanded as:-

- Addressing arbitrary user movement patterns
  - Time-varying user movement rates
  - Unrestricted non-random user movement, and
- Addressing arbitrary network topologies
  - Network boundaries
  - Overlapping cells
  - Varying cell sizes and average residence times
  - Directional network sections, such as highways and tunnels

A per-user dynamic location management scheme is designed in response to these goals. The proposal not only removes the restrictions on user characteristics and network topology enforced by traditional approaches, but capitalises on knowledge of these new characteristics, better tailoring an LM system to both user and network.

A location management scheme capable of addressing the requirements of a real communication network requires a radical shift in design philosophy away from characterising networks in terms of rigid physical parameters and high computational complexity. This chapter presents a location management scheme based on a simplified form of machine learning, able to adapt to an extremely wide variety of network topologies and user characteristics.

3.1 Graph Abstraction of Network Topology

A network representation based on physical coordinates provides a very restricted and overly complex description of a mobile network. While it is theoretically possible to describe a real network in terms of physical location, radius and shape of each individual cell, this system scales very poorly and is not able to be dynamically processed by a network. Traditional descriptions of such networks, based on idealised hexagonal or circular cells as described in Section 2.4.1.2, do not begin to describe the complex size and dispersion patterns possessed by real network cells. Additionally, despite the difficulties in describing such physical features, this information holds little semantic value. The mean time spent by users in a cell is not simply a function of cell radius, as evidenced by long dwell times for small cells covering an office building. Moreover the movements between cells are not simply a function of geographical coordinates, rather they are influenced strongly by cell directionality, physical obstructions and paths such as roads or highways.

A more intuitive and highly representative model of a cellular network is as a weighted directed graph. Here each node represents a cell; each keeping a record of the average time spent residing inside it by users. The weighted directed links represent the probability of movement from a given cell to each neighbour. These two simple quantities are able to fully describe all the relevant characteristics of a network topology for the purposes of location management, while being extremely lightweight in their description. An extremely simple example of such

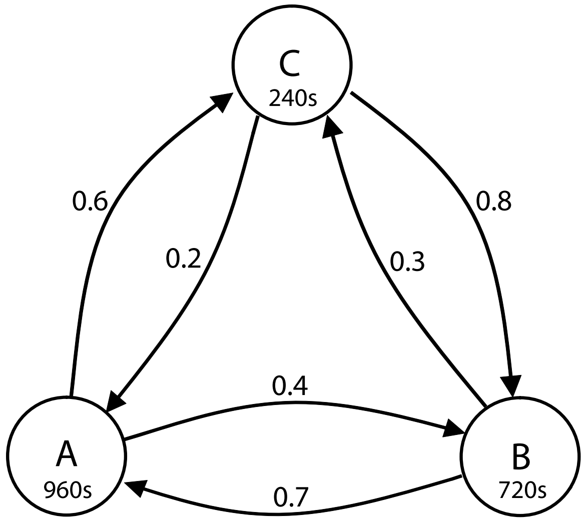


Figure 3.1: Graph topology representation of a highly simplified network

a graph topology is given in Figure 3.1 illustrating probabilities of user movement and cell dwell times.

The real beauty of an abstract graph network description however, is in its ability to dynamically adjust to capture changing characteristics of a network. In terms of reduced overhead, this allows the network to 'learn' its own topology and characteristics, relieving network operators from manually configuring such information. It also allows for additional cells to be installed or removed at will, with the network dynamically adjusting to reflect any physical changes made. From the perspective of accuracy such a scheme offers huge gains over traditional network descriptors. Rather than requiring a manual static description of the network, such an abstraction is able to adjust to capture temporally variant network characteristics in response to these network events themselves, ensuring the network parameterisation is optimised per time and day.

The details of the two abstract parameters of movement probability and dwell time are described following. Together they form a consistent and highly accurate view of the network, providing a foundation for the computation of location areas for individual users.

3.1.1 Movement Probabilities

Previous approaches to location management have assumed an equal priority of moving from a given cell to each of its neighbours. Movement predictions have instead been based on projected physical movement velocities of users. Not only is this approach computationally intractable for networks containing millions of active users, it is also a poor reflection of real user cell transitions. As shown in [CLSS00], cellular transitions for users with even highly directed physical movement are significantly random in their appearance, being highly decoupled from actual physical movements.

Far more influential on cell transitions are aggregate movement patterns, such as those resulting from users consistently moving along a stretch of highway. Here this movement is far easier to predict to an accurate level, and owing to the simplicity of their description, highly tractable in terms of feasible network implementation. Instead of attempting to predict the highly complex and variable movements of individual users, this scheme instead focuses on more established consistent characteristics of network movement, resulting from the physical topology surrounding each cell. Predictions for user movement can then be made from this aggregate model, in combination with per-user characteristics of movement and call *rate*, as described in Section 3.3.

3.1.1.1 Probability Calculation

Movement probabilities are calculated per-cell to each of its neighbours. While ideally these probabilities would encompass all movements out of a given cell, this would require a large amount of communication overhead, negating the benefits of an efficient location management scheme. Instead each cell records movement probabilities based on the frequency of handoffs to other cells, that is, cell transitions instigated when a user is on a call. As each cell must already be fully aware of a cell transition when a user is on a call, to correctly instigate a handoff, no additional communication overhead is required in recording these probabilities.

Probabilities are updated at specified *refresh intervals*, scheduled as necessary in the network. If a fine level of granularity is required, each cell may be refreshed hourly. If instead only low granularity is necessary, or the number of handoffs out of a cell is comparatively low, refreshing may take place daily or even more infrequently, once the network has reached an aggregate steady state.

The management of these probabilities may be computed at each *Base Station Controller*, or more centrally, collocated at the *HLR* to minimise required modifications to current networks. A general algorithm for the calculation of these movement probabilities per cell is given as follows:

1. For every handoff to a neighbour cell, increment the handoff tally for that particular neighbour, or record the cell as a neighbour if not previously seen
2. On a cell *refresh*, divide each individual handoff tally by the total number of handoffs to obtain the probability of moving to each cell in the last handoff interval
3. Update the stored movement probability to each cell, smoothing the values using a simple exponentially weighted moving average
4. Normalise the total probabilities of moving to each cell to ensure they sum to 1.0
5. Reset each handoff tally to zero and continue the process

As can be seen, the above algorithm is quite simple, requiring only two hash maps for implementation - one to store the handoff tallies in each refresh interval, and one to store the *smoothed* movement probabilities to each cell. As the number of possible neighbours for a cell is quite small, these data structures consume a minimum of storage space, and the probabilities require very little processing power to compute.

The probability smoothing is conducted to ensure an anomalous number of handoffs in a given cell refresh interval do not unnecessarily skew the results. This is performed in simple fashion as in Equation 3.1:

$$prob\left(c, i\right) = k \cdot prob\left(c, i-1\right) + \left(1-k\right) \cdot newProb\left(c\right) \tag{3.1}$$

for each cell c, smoothing factor k and recently recorded probability *newProb*. A smoothing factor, k, of 0 will ignore all previous values and store each new probability directly, to be used in very active networks with consistent probabilities. A network with fewer users however would benefit from a higher value of k, assisting in keeping the network in a state of equilibrium. $prob\,(c, 0)$ will be set at 0 initially, as no movement history will be held, yet within a small number of refresh cycles the probabilities will approach the mean value seen by each cell.

The smoothing of each probability results in a set of movement probabilities out of a cell that may not sum exactly to 1. A simple normalisation of these values at the end of the refresh cycle ensures that the recorded movement probabilities all sum to 1 and are probabilistically valid.

3.1.1.2 Pattern Recognition

After a series of repeated cell refreshes in a consistent network, user movement patterns will emerge. These patterns are temporal in nature, related to movement trends in the network. For example, in morning peak-hour the probability of a user moving south across the Sydney Harbour Bridge towards the Sydney CBD is much greater than that of a user moving north-bound, as will be reflected in morning cell movement probabilities. Conversely, the probability of a user moving north on the Harbour Bridge is much greater in the late afternoon. While these patterns will be well captured by a regularly refreshed cell set, opportunity exists to learn and cache these movement patterns, to avoid the requirement for regularly scheduled refresh events. This is left as an exercise for further work, owing primarily to the low computational overhead required in refreshing a cell, as compared to the requirements for temporal pattern matching and recollection.

3.1.2 Cell Dwell Times

The computation of dwell times for each cell is again performed on an aggregate basis, with each cell storing the average time spent by users when visiting the cell. This is an important characteristic regardless of individual user movement rates - the dwell time gives a measure of how 'large' a cell is, whether it is a physically large rural cell or merely a small urban cell through which users are forced to move slowly, and hence conceptually 'large'. The time spent within a cell by a user moving at a given movement rate will hence rely heavily on the average dwell time of users within the cell, more appropriately replacing traditional concepts of cell size. This average time is later transformed according to the movement rate of a given user when calculating location area size on a per-user dynamic basis.

3.1.2.1 Dwell Time Calculation

No refresh cycle is required in calculating dwell times, rather just smoothing of each recorded time, such as in Equation 3.1 with:

$$dwell\,(i) = j \cdot dwell\,(i-1) + (1-j) \cdot newDwell \tag{3.2}$$

Here the smoothing factor j would likely be higher than k, to average the effects of users with widely varying velocities moving through the cell. The value of $dwell\,(0)$ may be initially set to an arbitrary default value, with the cell quickly learning its own characteristics in response to recorded dwell times.

The update of dwell times clearly requires knowledge of the time spent by each user within the cell. This may be obtained in one of two ways:

- Recorded when a user moves into a cell while on a call, via handoff, then hands off again when leaving the cell still on the call
- Reported directly by a device when it moves out of the cell while on a call, at the same time as the cell is updating its handoff tally in response to the movement

The former method requires no additional communication overhead yet offers poor granularity. In networks with low levels of call activity the proportion of cell visits that begin and end with a user on the same call will be very low. It also introduces a bias against long dwell times, since it is far less likely that a user remaining within a cell for a long period of time will be on a call for this entire duration. The latter method offers far greater accuracy, at the expense of a small communication overhead. Here the device is required to inform a cell of its own dwell time as it leaves on a handoff. Since the device already has a record of this dwell time, as will be described in Section 3.3, and must already initiate a communication session with the cell to perform handoff, this overhead is comparatively small. The dwell time may be 'piggybacked' on the existing handoff packets, increasing communication overhead by only a few bytes. Here there is no bias against length of a user's stay within a cell, and the granularity offered in recording this information is conveniently equal to that used in calculating movement probabilities.

3.1.2.2 Pattern Recognition

As with the movement probability metric, a degree of pattern matching is again possible with dwell times at each cell. Such a scheme would be able to cache regular occurrences such as an increase in dwell time for highway cells during peak hour. Such a scheme would offer little reduction in overhead however, since user devices would still transmit dwell time information on handoffs, regardless of whether the cell has leant its temporal characteristics of dwell times. Once this time information is available from the user, little further computation is required; a reduction in this not justifying the increase in complexity required by a pattern matching engine.

3.2 LA Insertion Order

While simplistic location management schemes are restricted to allocating circular location areas to users, a scheme with knowledge of network movement probabilities is able to make a more informed selection when determining the shape of a given location area. Such a scheme may conform to the logical structure of cells in a network, able to assign an elongated location area along a stretch of highway, or bias the shape of a location area towards directions of common user movement. This flexibility in location area assignation offers significant reductions in location management cost - here the number of cells in a location area may be minimised while ensuring inclusion of all cells likely to be visited by the user. These gains hinge solely on the description of a network in terms of a weighted graph for efficient implementation, as presented in Section 3.1.

The informed assignation of an irregular location area requires knowledge of the most likely cells for a user to visit. Here a prioritisation scheme is used, ordering all neighbouring cells by their probability of being visited by a user. This establishes the precise order that cells should be added to a location area to minimise the chance of leaving the LA for a given number of cells. As movement probabilities are calculated on an aggregate basis, this ordering may be computed once per cell refresh cycle, cached in an array used whenever a location area need be determined for a user.

3.2.1 Cell Ordering

The cell ordering is computed based on the movement probabilities recorded for each cell, and most efficiently performed by a central network controller. A complete network map recording the most recent movement probabilities between each cell is sufficient to compute the location area cell insertion order for every cell in the network. This order is discovered via a uniform cost search through a cell's neighbours, based on the probability of a user moving down a given network path.

By adding each cell in decreasing probability of being visited by a user, we ensure that for a given number of cells in the LA, the user will have the greatest chance of remaining within those cells for an extended period of time. Here shapes and arrangements of location areas are not specifically considered, rather allowing the mathematical probabilities to implicitly define these characteristics. This provides an ideal selection of cells for each location area, without the overhead of complex heuristics attempting to manually determine location area shape.

The basic procedure for calculating cell insertion order, recomputed after each cell refresh cycle, is given in Algorithm 1. An example will follow in Section 3.2.2.

Algorithm 1 Procedure for determining LA cell insertion order for a given cell location

```
initialise priority queue searchQueue
initialise array LAOrder, used to store the order of LA cell inser-
tions

insert current cell in searchQueue, with probability 1.0

while(termination criteria not reached) {
    pop cell c with probability p from top of searchQueue
    add cell c to LAOrder if not already present

    for(all neighbours n of cell c) {
        initialise probability q of user visit-
        ing cell n along current path
        read movement probability m from cell c to cell n

        q = p * m

        if(q is above a minimum probability threshold) {
            insert cell n in searchQueue with probabil-
            ity q
        }
    }
}
```

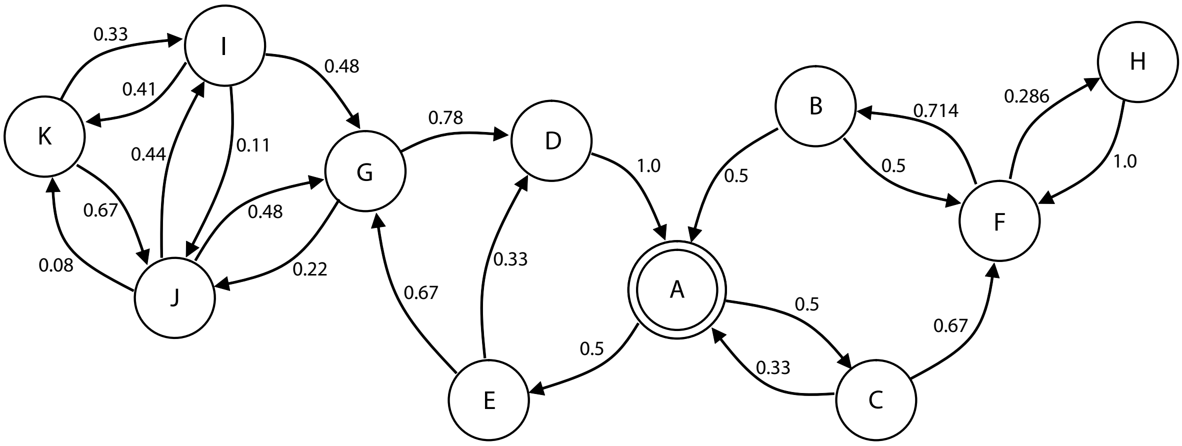


Figure 3.2: Partially 'learnt' network region, generated from SUMATRA BALI-2 [PLE04] network topology.

Note that repeated cell visits are not pruned from the search tree as they are encountered while expanding path priorities. While repeatedly occurring cells may slow the rate of insertion of cells into the LA Order array, these are an important characteristic of real user movement. Only by considering the common occurrence of users repeatedly moving between a small group of cells, can an accurate record of path likelihood be established.

A discussion of the termination criteria used to halt the search will be left for Section 3.2.3. The following section presents an example of the cell ordering process, given to establish a concrete implementation of Algorithm 1.

3.2.2 LA Order Example

Table 3.2 shows the partial execution through Algorithm 1, for the network segment illustrated in Figure 3.2. This network topology is the result of a small number of refresh cycles executed on the SUMATRA BALI-2 trace data discussed in Section 4.1. Here the LA insertion order is being determined for users performing a location update at cell A.

After 10 search steps the search the network has determined the order of first 7 cells that should be added to a location area for a user updating at cell A. If the ideal LA size is determined to be 3 cells for a given user, via cost function minimisation as in Section 3.6,

Search Queue	Active Cell	LA Order
[A:1.0]	[A:1.0]	A
[C:0.5*1.0=0.5], [E:0.5*1.0=0.5]	[C:0.5]	A, C
[E:0.5], [F:0.67*0.5=0.335], [A:0.33*0.5=0.165]	[E:0.5]	A, C, E
[F:0.335], [G:0.67*0.5=0.335], [A:0.165], [D:0.33*0.5=0.165]	[F:0.355]	A, C, E ,F
[G:0.335], [B:0.714*0.355=0.253], [A:0.165], [D:0.165], [H:0.286*0.355=0.102]	[G:0.335]	A, C, E, F, G
[D:0.78*0.335=0.261], [B:0.253], [A:0.165], [D:0.165], [H:0.102], [J:0.22*0.335=0.074]	[D:0.261]	A, C, E, F, G, D
[A:1.0*0.261=0.261], [B:0.253], [A:0.165], [D:0.165], [H:0.102], [J:0.074]	[A:0.261]	A, C, E, F, G, D
[B:0.253], [A:0.165], [D:0.165], [E:0.5*0.261=0.131], [C:0.5*0.261=0.131], [H:0.102], [J:0.074]	[B:0.253]	A, C, E, F, G, D, B
[A:0.165], [D:0.165], [E:0.131], [A:0.5*0.253=0.127], [F:0.5*0.253=0.127], [C:0.131], [H:0.102], [J:0.074]	[A:0.165]	A, C, E, F, G, D, B
...	...	...

Table 3.2: Development of LA Order array for network given in Figure 3.8

then only the first three cells from the LA Order list will be assigned to the user, in this case the location area $\{A, C, E\}$. While cells B and D are in similar proximity to cell A as are C and E, they are far less likely to be visited by a user, as reflected by their position further down the LA Order list. This subtlety is easily captured by the cell ordering algorithm presented here, yet much more difficult using traditional physical based LA assignation.

3.2.3 Termination Criteria

The cell expansion process may be continued until:

1. The Active Cell probability decreases below a pre-set minimum threshold
2. The number of cells in the LA Order array reaches an optional maximal size
3. All cells in the network have been added to the LA Order array.

Each termination criterion is introduced as a result of a physical limitation in the network. The minimum probability threshold is effectively a restriction on the number of iterations performed in the search queue expansion, to limit processing time. Rather than a hard limit on an exact number of iterations, this method allows the specification of a required confidence level. A minimum probability threshold of 0.001 will ensure that all paths with probability greater than 0.1% will be considered in determining the LA insertion order. Cells need not be added to the priority queue if they are themselves below the threshold, preventing the priority queue from increasing in size indefinitely.

A restriction on LA size is required only in extreme cases of a very large network or tight storage requirements. This metric is far less reliable than that described above, since in networks with high probability variance there may be little correlation between search size and the size of the LA Order array. As the size of an assigned LA is dictated by the movement probabilities between cells, rather than the number of cells themselves (see Section 3.5), it is not possible to determine a 'sensible' size threshold without prior knowledge of movement probabilities.

Termination when all cells in the network have been added is the ideal case, yet infeasible for extremely large networks. Here the LA Order array will be sufficiently large to accommodate for any possible location area size. In most cases this criteria will never be reached, owing to the excessive computation involved and the highly infrequent case that a location area need extend over the entirety of a mobile network.

3.3 User Characterisation

The size of an optimal location area is heavily dependent on the characteristics of the user performing the location update. A user receiving a large number of calls will place a large paging burden on the network, while a user moving rapidly will likely trigger a large number of location updates. A *per-user* dynamic LM must hence characterise each user in terms of their call arrival and movement rates, using this information to generate an appropriate location area in minimising cost.

Previous research has attempted to additionally characterise users in terms of their precise

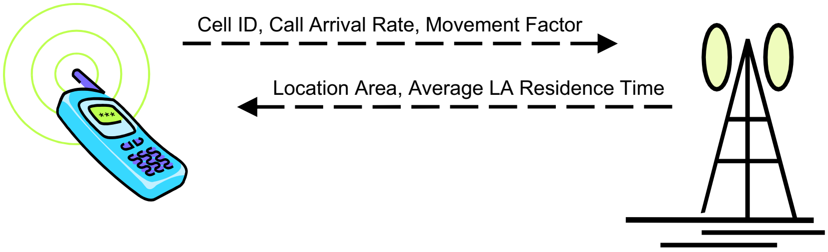


Figure 3.3: Communications involved in a location update

movement patterns, assigning movement vectors to each user in the network [LR98, HCT00, CBCS02]. It may be noted from the previous section that in the proposed scheme movement patterns are characterised purely on an aggregate basis, to capture large-scale movement characteristics at each cell, rather than on a per-user basis. The reasons for this decision are two-fold:-

- the concept of dynamically characterising the movements of millions of individual users via complex mathematical constructs and heuristics goes strongly against the goal of implementation feasibility;
- as shown in [CLSS00], the cell movements of users in dense heterogeneous networks are highly unpredictable and markedly detached from physical movement vectors, circumventing efforts to accurately predict future location from the past movements.

The communication of user parameters to the network is performed on a location update, transmitting call and movement rate parameters to the network along with the cell ID information already required as part of the location update protocol. The network is able to generate an optimal location area for the user based on this information only, without recording any state information on each user. This communication is summarised in Figure 3.3.

The proposed user characterisation scheme, comprising a call arrival rate and movement factor per user, is sufficiently lightweight that it may be implemented purely within currently available cellular phones. A granularity of 2 bytes per parameter would occupy only 4 bytes of space on a device's SIM card, with the simple multiplication and addition of 2 byte values on a call receive event or location update well within the computational power of even a very primitive mobile device. The precise calculations of these quantities are given in the following sections.

3.3.1 Call Rate Estimation

The estimation of call arrival rate λ is relatively straightforward and consistent amongst the great majority of per-user dynamic location management schemes. Here the user actually maintains a record of the interval between received calls, converting this to a calls-per-hour rate as required on a location update. Note that the interval is between *received calls*, not simply calls to or from a device. As no paging is required to find a user when they are making a call, only the rate of received calls influences the LM cost for the user.

Call interval is recorded via a timer set at each call receive. Each interval is smoothed with relation to the previously recorded intervals to reduce the effect of anomalous call events. This is shown performed with smoothing factor g in Equation 3.3. The value of $callInterval\,(0)$ must be set to an appropriate median value, approaching the real value for the user after a number of calls have been recorded.

$$callInterval\,(i) = g \cdot callInterval\,(i-1) + (1-g) \cdot newCallInterval \tag{3.3}$$

The calculation of the call arrival rate to be transmitted to the network on a location update is just an inversion of the call interval as in Equation 3.4. The factor 3600 here is used simply in converting the units of call rate from calls/second to a more easily represented calls/hour.

$$\lambda = \frac{3600}{callInterval} \tag{3.4}$$

3.3.2 Movement Factor Estimation

Traditionally movement characterisation has been quantified in terms of linear velocity or cell-crossings per second. While this applies to homogenous cellular networks where cell sizes are identical and dwell time within each cell is equal, such assumptions are invalid for realistic heterogeneous networks. Regardless of physical movement rate a user will spend longer in a large cell than a small cell, with cell dwell time also affected by the number of overlapping cells in the area and physical geography surrounding the cell. It thus makes little sense to characterise user movement in terms of absolute quantities.

Here a more appropriate *relative* movement concept is maintained - that of user movement *factor* γ. This factor determines how fast a user moves through a given area in relation to the average user, and may be determined for a specified region as:

$$\gamma = \frac{\text{time taken for average user to move through region}}{\text{time taken to move through region}} \tag{3.5}$$

An extremely important side effect of such a parameterisation is that users may make an infinite number of 'ping-ponging' hops between a pair of cells within the same region, without affecting their movement factor. This is a very useful characteristic - it abstracts movement based purely on network characteristics, such as hopping between neighbouring cells, from genuine user movement over a larger area. While previous proposals based on individual cell crossings become increasingly irrelevant as network density and hence ping-pong-like behaviour increases, the movement factor metric retains its accuracy.

With an accurate record of movement factor sent on a location update, the network is able to estimate the time spent within a given LA for this specific user. What remains however is how to determine the time taken for the *average* user to move through an identical region as the user calculating movement factor.

The one occasion when a user device knows exactly the average time taken for an average user to move through a certain region is on a location update. Here, as detailed in Section 3.5, the network must already estimate $\widehat{T_{LA}}$ the time taken for the average user to leave a specified location area. When the network sends the specified location area to the user, it also sends this value of $\widehat{T_{LA}}$, informing the device of this required quantity. Upon leaving this new location area the device may compare how long the network expected the average user to reside within the LA, with how long the device actually resided within the LA, to determine how 'fast' the user is moving compared to the network's view of an average user.

The user must hence start a *residenceTime* timer after each location update, the value of which is used to estimate the new movement factor at the next update. This is computed based on the previous value of $\widehat{T_{LA}}$ as:

$$new\gamma = \frac{\widehat{T_{LA}}}{residenceTime} \tag{3.6}$$

Again this value is smoothed to remove any artefacts caused by an isolated burst of uncharacteristic movement. This is computed based on the previous movement factor value and smoothing factor h in Equation 3.7. $\gamma(0)$ is defined to be 1, treating all users as equal until their movement has been observed.

$$\gamma(i) = h \cdot \gamma(i-1) + (1-h) \cdot new\gamma \tag{3.7}$$

While the low granularity of estimating user velocity once per location update seems initially to be insufficient, it is important to recognise that the only time when a profile of user movement is required by the network is precisely on these location updates. As the user will be expected to reside within a location area for an extended period of time, it is misleading to base predictions of residence time on instantaneous velocity at a location update. Instead it is far more suitable to characterise user movement on a granularity matching that required by the network, capturing an estimation of long-term user movement rate.

3.4 Cost Function Formulation

One of the more significant developments comprising the location management scheme is the formulation of a cost function, used to determine the ideal location area size based on both network and user characteristics. Such a function provides the network with a quantitative measure of the estimated cost that will be incurred when assigning a given location area to a user. With this cost function minimised, as in Section 3.6, the network can ensure the minimum total cost incurred, and hence the most efficient LM implementation.

The total location management cost for a network may be evaluated as the combined costs of location updates, as well as the cost of sending paging messages to locate a user when they are to receive a call. These are the only two occasions when a quantifiable cost is incurred on the network as a result of managing the location of users. Small location areas will lead to a low paging cost, since the location of the user is known to relatively high accuracy, but high location update cost, as the user will likely leave their location area after a very small period of time. Conversely a large location area will minimise location update cost but increase the paging load when the user receives a call. The very art of location management is in minimising this aggregate cost in response to a set of network and user parameters.

The total location management cost per hour, for a given location area, may be expressed formulaically, as in [Kol03] as:

$$\text{Total Cost} = \text{Location Update Cost} + \text{Paging Cost}$$

$$C_{TOTAL} = C_{LU} + C_P \tag{3.8}$$

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_{PAGING} terms is dependent however on specific LM implementations.

The paging cost for our dynamic location management scheme may be estimated simply based on a notion of the average call receive rate for a user, along with the number of cells in the location area. The incoming call rate is provided to the network by the user on a location update while the exact number of cells in a given location area is known by its very definition. Together these two parameters will provide us with an estimate of the number of individual pages that will be conducted per hour within the location area. An additional parameter ς_P is required to define the cost of each individual paging message, related to paging cost in bytes and resource penalties enforced on the network. The size of this factor and its relation to location update cost is explored in Section 3.7. The paging cost for a given location area may be given, again as in [Kol03], as Equation 3.9, measured here in dollars per hour.

$$\text{Paging Cost} = \#\text{incoming calls per hour} \times \#\text{cells in paging area}$$

$$C_P = \lambda \cdot \varsigma_P \cdot N_C \tag{3.9}$$

The calculation of location updating cost requires a more complex level of analysis. As with paging, the location update cost will be related to the cost per update ς_{LU}, a factor that will be examined in Section 3.7. What is hence required is an estimation of the number of location updates a user will perform per hour, given their current location and user characteristics. This calculation is relatively straightforward in a homogenous network, assuming completely random movement over a perfectly tessellated network of hexagonal cells. Here the user's movement rate μ may be divided by the total number of moves P_{LA} they will likely make before leaving a given location area, to estimate how long the user will spend in the current LA. [Kol03] represents this computation as:

$$\text{Location Update Cost} = \left(\frac{\text{user movement rate}}{\text{est. moves to leave LA}}\right) \times \text{cost per LU}$$

$$C_{LU} = \frac{\mu}{P_{LA}} \cdot \varsigma_{LU} \tag{3.10}$$

The calculation of P_{LA} in a homogeneous network can be efficiently implemented via a lookup table indexing movement counts for various location area radii. In a heterogeneous network however, such concepts of LA radius do not apply. Moreover each location area may be of arbitrary shape and complexity, prohibiting static lookup of such movement values. Here the $\frac{\mu}{P_{LA}}$ factor is replaced by $\frac{1}{T_{LA}}$, representing the inverse of the estimated time a specific user will spend within the location area. The additional level of abstraction provided by T_{LA} allows a flexible approach to location management - offering a formulation of location management cost able to be applied to a wide variety of present and future LM schemes.

The reformulated generalised representation of location update cost is hence:

$$\text{Location Update Cost} = \frac{\text{cost per LU}}{\text{est. residence time within current LA}}$$

$$C_{LU} = \frac{\varsigma_{LU}}{T_{LA}} \tag{3.11}$$

Substituting the above into Equation 3.8, the complete general formula for total location management cost for a given user and location area is:

$$C_{TOTAL} = \frac{\varsigma_{LU}}{T_{LA}} + \lambda \cdot \varsigma_P \cdot N_C \tag{3.12}$$

The precise calculation of T_{LA} for the proposed scheme, the estimated residence time for a user within the LA, will be detailed in the following section.

3.5 Estimation of T_{LA}

With the cost function for a location area redefined to depend on T_{LA}, the average time spent by a user in a given location area, this estimation becomes central to the performance of a dynamic location management scheme capable of operating in heterogeneous networks. While the mean dwell time for each cell within the location area is known, in addition to the average movement rate of the user, this information alone is not sufficient to accurately estimate the time the user will spend within the LA. Complex factors related to the shape and size of the LA, as well as regions where users repeatedly 'ping-pong' between a small group of cells, have a highly influential effect on the time it will take for the user to leave their location area. Fortunately this information is captured completely and with generality by the directed weighted graph network topology.

If the network graph of a location area is viewed as a state machine, with the current cell representing the start state, and all neighbours outside the LA representing 'sinks' or end states, then the average time taken to complete a pass through this state machine will give us the average time spent by a user within the location area itself. Here the dwell time at a cell represents the average 'processing time' at a state, while cell movement probabilities define the likelihood of moving between given states. An example of such a state machine can be seen in Figure 3.4, representing the three states within a small location area, starting at cell A.

3.5.1 Calculation of State Machine Execution Time

The time spent within the location area state machine may be calculated by solving a probability matrix, determining the average number of visits made to each state before exiting. Once the number of visits at each cell has been determined, these may be multiplied by their corresponding cell dwell times and summed to determine the mean time spent in the entire location area.

First a set of equations defining the probable number of visits p_S to each state S must be established. The probability of visiting a given state is equal to the sum of the number of times each of its neighbours is visited, each multiplied by the probability of moving from the neighbour to the cell itself. While best illustrated via an example, as in Section 3.5.2, the general form will be given for reference below.

The set of equations for an LA containing cells $\{S_1, S_2, \ldots S_N\}$, with current cell S_1 are given as:

$$\begin{array}{lcllll}
p_{S_1} & = & 1 & & +P\left(S_2,S_1\right)\cdot p_{S_2} & \cdots & +P\left(S_N,S_1\right)\cdot p_{S_N} \\
p_{S_2} & = & & P\left(S_1,S_2\right)\cdot p_{S_1} & & \cdots & +P\left(S_N,S_2\right)\cdot p_{S_N} \\
\vdots & & & & & & \\
p_{S_N} & = & & P\left(S_1,S_N\right)\cdot p_{S_1} & +P\left(S_2,S_N\right)\cdot p_{S_2} & \cdots &
\end{array} \tag{3.13}$$

Here in Equation 3.13 $P(Si,Sj)$ represents the probability of moving from cell i to cell j, as learnt by the network in Section 3.1.1.1. p_{S_1} will always have 1 added to it before any other probabilities are summed, since the start state by definition is visited at least once regardless of movements throughout the state machine. More conveniently represented in matrix notation, the system of equations becomes:

$$\begin{bmatrix} 1 & -P\left(S_2,S_1\right) & \cdots & -P\left(S_N,S_1\right) \\ -P\left(S_1,S_2\right) & 1 & \cdots & -P\left(S_N,S_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ -P\left(S_1,S_N\right) & -P\left(S_2,S_N\right) & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_{S_1} \\ p_{S_2} \\ \vdots \\ p_{S_N} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.14}$$

In a real network implementation this matrix will be quite sparse, as $P(Si,Sj)$, the probability of moving directly between two arbitrary cells i and j, is zero unless they are neighbours. This provides significant performance gains when solving the matrix.

With values inserted for each $P(Si,Sj)$, the matrix equation may be solved to determine the values of $p_{S_1},\ldots,p_{S_N}$. If the matrix is singular, and hence cannot be solved, it indicates that the probability of leaving the state machine is zero, and hence the user will never leave the location area. In this special case, such as when the location area covers the entire network, T_{LA} is infinite and subsequently the location updating cost component C_{LU} is zero.

Together with D_{S_i}, the mean dwell time at each cell S_i, the time $\widehat{T_{LA}}$ spent within the location area for an *average* user may be computed as:

$$\widehat{T_{LA}} = D_{S_1}\cdot p_{S_1} + D_{S_2}\cdot p_{S_2} + D_{S_N}\cdot p_{S_N} \tag{3.15}$$

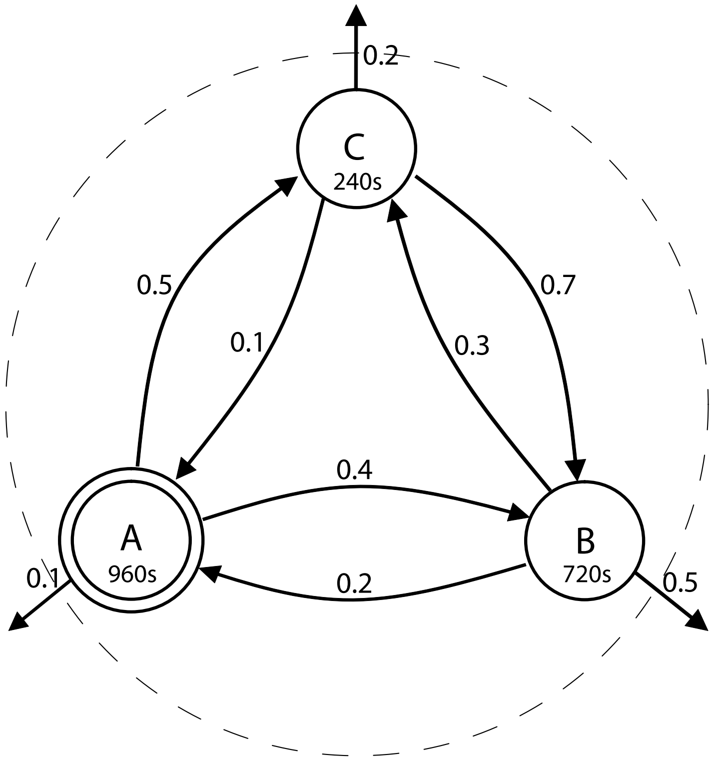


Figure 3.4: State machine interpretation of a location area

Estimating the time spent by a specific user within this location area is then a simple matter of dividing this average residence time by the user's movement factor γ to determine the final estimated residence time:

$$T_{LA} = \frac{\widehat{T_{LA}}}{\gamma} \tag{3.16}$$

3.5.2 LA Residence Time Example

A simplified example of the computation of LA Residence time is given here, to clarify the theory presented in the previous section. The estimation of average LA residence time T_{LA} for the *average* user is performed for the simple location area of Figure 3.4, presented in state machine format. Here the mean dwell times for cells A, B and C are 960, 720 and 240 seconds respectively, as shown within each 'state' in the figure. The start state, A, is the cell within which the user performs the location update, and all cells outside the dotted location area are considered terminal states for the system.

Following the logic of Section 3.5.1, the equations representing the average number of times each state will be visited are:

$$\begin{aligned} p_A &= 1 \quad +0.2 \cdot p_B \quad +0.1 \cdot p_C \\ p_B &= \quad 0.4 \cdot p_A \quad +0.7 \cdot p_C \\ p_C &= \quad 0.5 \cdot p_A \quad +0.3 \cdot p_B \end{aligned}$$

These may be interpreted, for example, that there is 0.4 probability that cell B will be visited after the state at cell A is active, and 0.7 probability it will be visited after the state is at cell C, as reflected in the movement probabilities of the topology. Rearranging and converting into matrix form we have the series of steps:

1.

$$\begin{array}{rclll} p_A &=& 1 & & +0.2 \cdot p_B \quad +0.1 \cdot p_C \\ p_B &=& & 0.4 \cdot p_A & \qquad\qquad\quad +0.7 \cdot p_C \\ p_C &=& & 0.5 \cdot p_A & +0.3 \cdot p_B \end{array}$$

2.

$$\begin{array}{rrrcl} p_A & -0.2 \cdot p_B & -0.1 \cdot p_C & = & 1 \\ -0.4 \cdot p_A & +p_B & -0.7 \cdot p_C & = & 0 \\ -0.5 \cdot p_A & -0.3 \cdot p_B & +p_C & = & 0 \end{array}$$

3.

$$\begin{bmatrix} 1 & -0.2 & -0.1 \\ -0.4 & 1 & -0.7 \\ -0.5 & -0.3 & 1 \end{bmatrix} \begin{bmatrix} p_A \\ p_B \\ p_C \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

This matrix equation may then be solved via Gaussian elimination or more complex and efficient methods as in Section 3.5.3, to obtain solutions:

$$
\begin{aligned}
p_A &= 1.3668 \\
p_B &= 1.2976 \\
p_C &= 1.0727
\end{aligned}
$$

Note that if the matrix were singular, and hence unable to be solved, we would assign an infinite value to T_{LA} as mentioned previously.

Combining these values with the dwell times at each cell we obtain the estimated residence time, for the average user, as:

$$
\begin{aligned}
\widehat{T_{LA}} &= D_A \cdot p_A + D_B \cdot p_B + D_C \cdot p_C \\
&= 320 \times 1.3668 + 240 \times 1.2976 + 80 \times 1.0727 \\
&= 834.6s
\end{aligned}
$$

Hence for an average user we have $T_{LA} = \frac{\widehat{T_{LA}}}{1} = 834.6s$. If the particular user was recorded as moving 3 times as fast as an average user, their estimated LA residence time would be $T_{LA} = \frac{\widehat{T_{LA}}}{3} = 278.2s$. This residence time can then be substituted back into the general LM cost equation, Equation 3.12, along with an LA size N_C of 3, to determine the estimated cost to the network for assigning the location area of Figure 3.4.

3.5.3 Matrix Computation Efficiency

An obvious concern in an implementation purported to provide high levels of computational efficiency is the presence of a seemingly inefficient matrix computation. The majority of mathematical operations required in implementing the location management scheme are simple multiplications or divisions, requiring little in the way of computational resources. The major iterative computation involved in determining LA cell insertion order also requires relatively little resources provided an efficient search queue implementation, and may be conducted infrequently if the network is running over capacity from a computational viewpoint. The matrix computation of Equation 3.14 must be performed at every location update however, without any degree of flexibility according on available computational power.

It is hence pertinent to ensure that the matrix computation involved in calculating residence time does not violate the goal of computational tractability, separating the proposed scheme from computationally intensive heuristic-based LM schemes. While the implemented simulation environment of Chapter 5 uses simple Gaussian elimination to solve Equation 3.14 for a given LA size, more efficient means may be utilised to reduce computational overhead. For a large location area this matrix will be highly sparse, owing to practical limitations on neighbourhood connectivity between cells in a network. As the majority of cells will not be direct neighbours of each other, the majority of array positions will be zero, owing to zero movement probability between two non-neighbour cells. Many techniques are available for efficiently solving systems of sparse matrices, with recent developments offering even greater performance gains [Gup02]. Additionally the evaluation of many similar location areas during the cost minimisation of Section 3.6 offers opportunity to 'reuse' portions of the calculation of a large location area in solving similar smaller LAs. While the examination of such mathematical optimisation techniques is out of the scope of this project, they do offer great opportunity for the reduction in computational cost.

As the matrix equation is used to determine the residence time for the *average* user in a given location area, in a completely optimal implementation this information need only be processed for once for each location area per refresh cycle. As each location update at a cell within a given refresh cycle utilises the same cell insertion order, there is potential for a greatly reduced number of calculations by caching the average residence time $\widehat{T_{LA}}$ for each increment in size of LA taken from this insertion order. The additional complexity in determining when cell insertion order or cell residence time have changed, in addition to the memory required in caching these values, justifies such optimisations only in a system with a great number of users, an hence high likelihood of repeated updates made in the same cell within a refresh cycle.

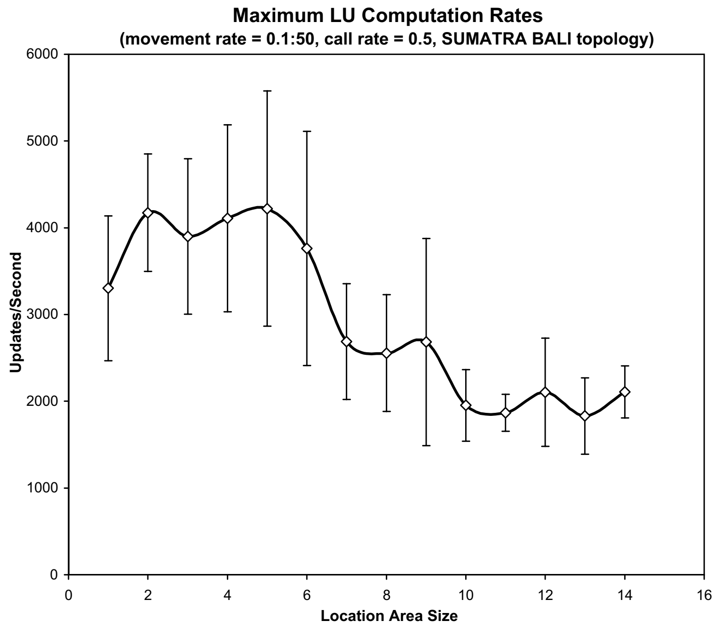


Figure 3.5: Location update rates available for typical network executions on consumer hardware

Without the implementation of sophisticated optimisation techniques, computational overhead is still low enough to allow relatively high location update rates on consumer hardware. Figure 3.5 shows the location update rates available when running a complete network simulation on a Pentium 4 3.0GHz PC. Here movement rates are varied from 0.1 moves/hour to 50 moves/hour to capture a wide variety of user behaviour. While there is significant variance in execution times the system is able to consistently process around 2000 location updates every second, on top of the regular simulation overhead. Note also that this simulation is not optimised for performance and hence much higher rates may be expected in a real network implementation. A fully optimised location management system, built as an implementation prototype rather than a complex simulation engine, is an area for future work. It is expected such an implementation would be capable of execution rates many orders of magnitude greater than those presented here.

3.6 Cost Minimisation

By Equation 3.9, as location area size N_C increases, total paging cost is monotonically increasing. Similarly, as new cells are added to a location area, estimated LA residence time T_{LA} also increases, and hence from Equation 3.11 location update cost is monotonically decreasing. As the cost function of Equation 3.8 defines location management cost simply as the sum of these two quantities, there must only be one minimum cost value for a given cell insertion order. The problem of cost minimisation then reduces simply to finding the local (and hence global) minimum for the cost function presented in Equation 3.12.

In the simulation implementation of Chapter 5 this minimum is found by a simplistic linear search through the function search space. Here the candidate location area starts as a single cell and each cell in the insertion order is added to the location area one-by-one, evaluating the cost function at each step, until there is an increase in cost over the previous value. At this point the previously added cell is removed and the cells remaining define the optimal location area for the given user. As the computational overhead involved in computing the cost for a small LA is very low, such a scheme works well for small to medium location areas, containing under 20 cells on average. From Figure 3.5 in the previous section it can be seen that this overhead is sufficiently small such that average location area size does not significantly affect system performance for small location areas.

For large location areas however a naïve linear search implementation contributes greatly to the total computation time required by the LM scheme. As this search must be performed at every location update, it is in the interests of a vendor to utilise an implementation that evaluates cost for a minimum number of candidate location areas during the search. The implementation of a *gradient-descent* search, using approximate gradients obtained from pairs of adjacent points in the search space, would be well justified for use in systems where large location areas may be expected. Here a small computational overhead is introduced in calculating the gradient at a point and estimating the location of the minima, yet the number of iterations through the search space would be greatly reduced. The nature of the cost function, with only one (global) minimum, opens the problem to a wide range of optimised search techniques.

Once the ideal minimum-cost location area is determined, the list of cells are sent to the client device using one of the representation methods of Chapter 6, completing the location update process. As presented in Figure 3.3 the data packet sent to the device contains not only the location area representation, but also the LA residence time estimation for an average user, defined in Section 3.5. The latter information is used by the device in determining its movement factor, as in Section 3.3.2.

3.7 $\varsigma_{LU} : \varsigma_P$ Ratio

Two critical components of the cost function not yet considered are the constants ς_{LU} and ς_P, defining the individual cost of a location update message and a paging message respectively. While not affecting the theoretical design of a location management scheme, they, or more specifically the $\varsigma_{LU} : \varsigma_P$ ratio, have a crucial effect on the expected cost for a given location area size. It is the reliance on the concept that location updating is a relatively expensive operation that makes significant performance gains possible by introducing the concept of location areas. An increased $\varsigma_{LU} : \varsigma_P$ ratio will favour a scheme with larger location areas, and hence less frequent updates at the expense of more paging messages, while a decreased ratio will have an effect to the contrary.

Without specific access to proprietary information on network packet formats and database loads, it is not possible to precisely quantify the cost relation between location updating and paging. Such information is clearly within access of network providers, who are free to set these ratios as appropriate in an implementation of the system. It is necessary however to establish an approximate value for this ratio to allow quantitative presentation of simulation results. This cost ratio is widely estimated to be roughly 10:1 [XTG93, Gon96]. With no evidence provided to the contrary, this 10:1 ratio will be used in the simulations of Chapter 5. The large static location areas in current GSM implementation, as investigated in Section 4.1, point to a ratio that may be significantly larger than this accepted value.

While previous research has quantified this ratio in bytes/hour [Kol03], the cost factors are redefined here purely in terms of dollars/hour. The implementation of comparatively complex location management schemes introduces additional computational expense in processing a location update, a factor which must be considered in the fair evaluation of such a proposal. As this cost cannot be captured merely in terms bytes send over the medium, it is far more appropriate to contrast net cost for location update and paging in terms of financial expense for the provider - reflecting the cost in occupying bandwidth, computational resources, and any additional infrastructure required. When such additional resources are considered, the comparative cost of location updating over paging in a dynamic LM scheme is almost certainly greater than the accepted 10:1 factor.

Fortunately the performance of the proposed scheme is relatively independent of specific $\varsigma_{LU} : \varsigma_P$ ratio. With suitable values inserted in the cost function of Equation 3.12 the cost will be minimised regardless of the relationship between paging and location update cost factors. In the extreme and highly unrealistic case of a 1:1 cost factor ratio, with a user with high call arrival rate, the system will simply assign the 'ideal' location area of a single cell. While the ideal location area is selected, the computational overhead involved precludes any advantages of a complex location management scheme in this case. With larger and more typical ratios of 10:1 or higher, location area size does not default simply to a single cell, and great performance gains are possible regardless of specific cost ratio.

3.8 Summary

This chapter has proposed a comprehensive location management scheme for modern cellular networks. This scheme is dynamic in that it assigns an ideal location area to a user based on not only the characteristics of their locality, but also the user's call arrival and movement rates. The scheme allows for heterogeneity by using a graph-based network topology abstraction, able to effectively describe an arbitrarily complex modern network, a unique contribution amongst current location management proposals. This abstraction is dynamic in itself, relieving network providers from manually configuring network parameters and able to capture temporally varying data on movement patterns throughout the network. A strong emphasis of this scheme has been on computational efficiency and feasibility of implementation, a focus also separating the proposal from much current location management research.

The following chapters will establish a benchmark for evaluating the performance of a location management proposal and verify the performance of this scheme under a variety of simulations.

Chapter 4

Benchmarking Network Performance

Before a performance investigation of a location management algorithm can be conducted, a firm concept of the precise performance requirements of modern networks is required. The great majority of previous research into location management has evaluated performance based purely on theoretical metrics and artificial traces, without comparison with existing communication systems. While such approaches allow comparison of individual implementations, albeit in a highly controlled homogenous environment, they offer no guarantees of specific performance in current and future systems.

This chapter will endeavour to evaluate the suitability of available network traces and establish a framework by which to effectively determine the performance of a location management scheme in a realistic network. As a focus on real-world performance comprises a central goal of this project, a thorough investigation will also be given into the characteristics of current implementations of GSM communication networks.

Once the suitability of network benchmarks is evaluated, a thorough test suite for location management schemes may be developed. As distanced from other research efforts, a knowledge of the relevance of each benchmark establishes a context by which to ultimately judge the performance of an LM algorithm, while assuring a degree of relevance in physical networks.

4.1 A Survey of Existing GSM Networks

While many novel and interesting technical developments have been proposed in the field of location management, little effort has been conducted into validating this research against real communication networks. With enormous evolution of mobile networks since early research into location management, it is important to re-establish a reference frame against which to examine the performance of a location management scheme.

In line with an emphasis on the practical real-world performance of an LM scheme, a comprehensive survey was conducted of current GSM communication networks, both locally and internationally. Over a period of three months cell movement data was recorded for a variety of user classes in major GSM networks of Sydney, Melbourne, London, Paris, Rome and Hong Kong. An analysis of this previously unavailable data allows direct validation of location management proposals, avoiding reliance on purely theoretical simulation. Such real trace data also allows an informed interpretation of the validity of theoretical simulation packages currently in use by the network research community.

4.1.1 Experimental Method

Cell logging software was written for mobile phones running the Symbian operating system [Ltd04], the details of which are given in Appendix A. This code ran as a background utility on a user's mobile phone, recording data on time, date, cell ID, location area code and cell name at every transition between cells in a GSM network. The latter cell name information was provided by the CellTrack utility for Nokia Series 60 devices [Fis04]. A sample of the type of data obtained using these utilities is given in Table 4.1.

Due to the limitations of requiring a relatively modern and expensive mobile phone to run such tracking software, the surveys were conducted over a user base of only a few individual users. The activities of these users were varied significantly over the long duration of the trials however, to encompass a wide range of user classes and behaviour. All trace information was recorded continually 24 hours a day to capture both day and night mobility behaviours.

Date	Time	Cell ID	Location Area Code	Cell Name
8/07/2007	23:48:55	10183	276	Epping
8/07/2007	23:49:16	671	276	Epping
8/07/2007	23:50:16	10581	277	Pennant Hll
8/07/2007	23:52:06	1332	277	Beecroft
8/07/2007	23:53:00	5463	277	Pennant Hll
8/07/2007	23:53:40	1331	277	Beecroft
8/07/2007	23:54:40	9501	277	Beecroft
8/07/2007	23:55:56	25162	277	Castle Hill
8/07/2007	23:57:34	6643	277	Dural
8/07/2007	23:57:43	25162	277	Castle Hill
8/07/2007	23:57:56	25161	277	Castle Hill
8/07/2007	23:59:28	6643	277	Dural

Table 4.1: Sample GSM location data

Significant variability in user behaviour and network location, over major networks in six cities as well as both urban and rural environments, ensured a broad and reasonably representative view of user movement behaviour in cellular networks. Call arrival probabilities however are not able to be captured simply by varying the behaviour and location of a small set of users. Such call rates are not easily described via segmentation into discrete user classes, and here no attempt was made to extrapolate average mobile phone call rates from such a small selection of users. Call frequency and distribution characteristics have been studied extensively in wired communication networks [CNLB92, Phi95], and hence these will be cast as a basis for representing call behaviour in a cellular network.

4.1.2 Network Topologies

While the majority of testing was conducted on the Telstra Sydney GSM network, trials were conducted on a total of 6 networks in 5 countries to ensure a wide and accurate representation of network characteristics. These networks, listed in Table 4.2, all operate under the GSM communications standard and were hence compatible with the cell tracking software developed for the local Sydney network. A summary of the individual network characteristics, along with a visualisation of each network topology as visited in the traces, is presented in Appendix B. Recorded trace duration was somewhat limited in the overseas networks although each trace

Country	City	Network
Australia	Sydney	Telstra MobileNet
Australia	Melbourne	Telstra MobileNet
England (UK)	London	Orange UK
Italy	Rome	Vodafone IT
France	Paris	Orange France
Hong Kong	Hong Kong	Orange HK

Table 4.2: GSM networks used in cell traces

City	Duration	Cells Visited	LAs Visited	Total Movements	Total LUs
Sydney	943 hours	437	9	11684	566
Melbourne	71 hours	109	5	1190	233
London	55 hours	276	18	1582	74
Rome	56 hours	218	5	1359	58
Paris	82 hours	345	15	1734	124
Hong Kong	54 hours	220	7	1285	44

Table 4.3: Trace characteristics of GSM networks

was run for at least 24 hours, with a high number of recorded movements, as demonstrated in Table 4.3.

Figure 4.1 compares the average movement and location update rates in each of these networks. While there is a degree of variance in these values, the distribution is correlated highly enough to make generalisations on *realistic* cell movement rates in current GSM networks. Notably, this allows an informed evaluation of mobility models in current use, as conducted in Section 4.2. There also appears to be a loose relation between network density and movement rates, as would be intuitively expected, supporting predictions that cell crossing rates will increase with further network development.

Importantly, the network traces allow the construction of a logical map of each GSM network, based on recorded cell transitions. While not comprehensive maps of the entire networks, these allow an insight into network structure and cell interrelations. Figure 4.2 shows such a map for a portion of the Vodafone IT Rome GSM network. While displaying only the cells encountered by the user during the trace, it clearly dispels previous assumptions of a uniform hexagonally tessellated cell structure, without network boundary or overlapping cells. Here

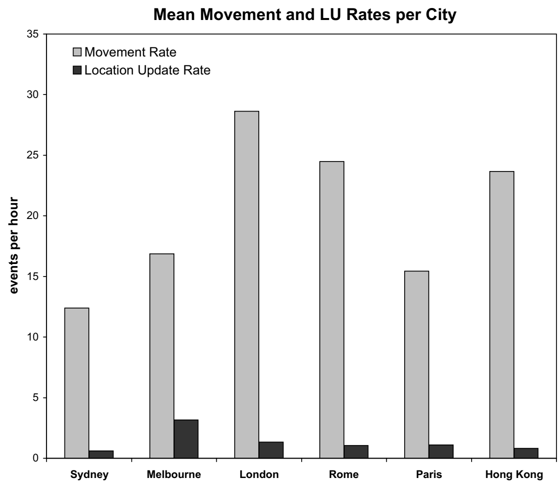


Figure 4.1: Mean movement and location updates recorded per city

dense regions in urban centres are visible as tightly packed regions of overlapping cells, with high connectivity, while sparse directed sections represent highways and major thoroughfares connecting such regions.

4.1.3 Network Characteristics

The average static location area sizes of the surveyed networks, displayed in Table 4.4, are significantly larger than those present in current network benchmarks such as SUMATRA, discussed in Section 4.2. It should be noted that the real static location area sizes may be considered to be much larger than that observed during the traces, since it is highly unlikely that every cell would be visited within a given LA during a trace run,leading to an underestimation of total cell count per observed location area code. The LA size difference between modern networks and traditional benchmarks may be justified by increased movement rate within the surveyed networks, along with a higher cellular density.

Notable from the traces are a number of discrete cells sharing each cell name, an example being the three cells `1332`, `1331` and `9501` described by the cell name 'Beecroft' in Figure 4.1. Such a network configuration lends evidence to the suggestion that each physical location is covered by a number of separate cells. This overlapping network configuration leads to

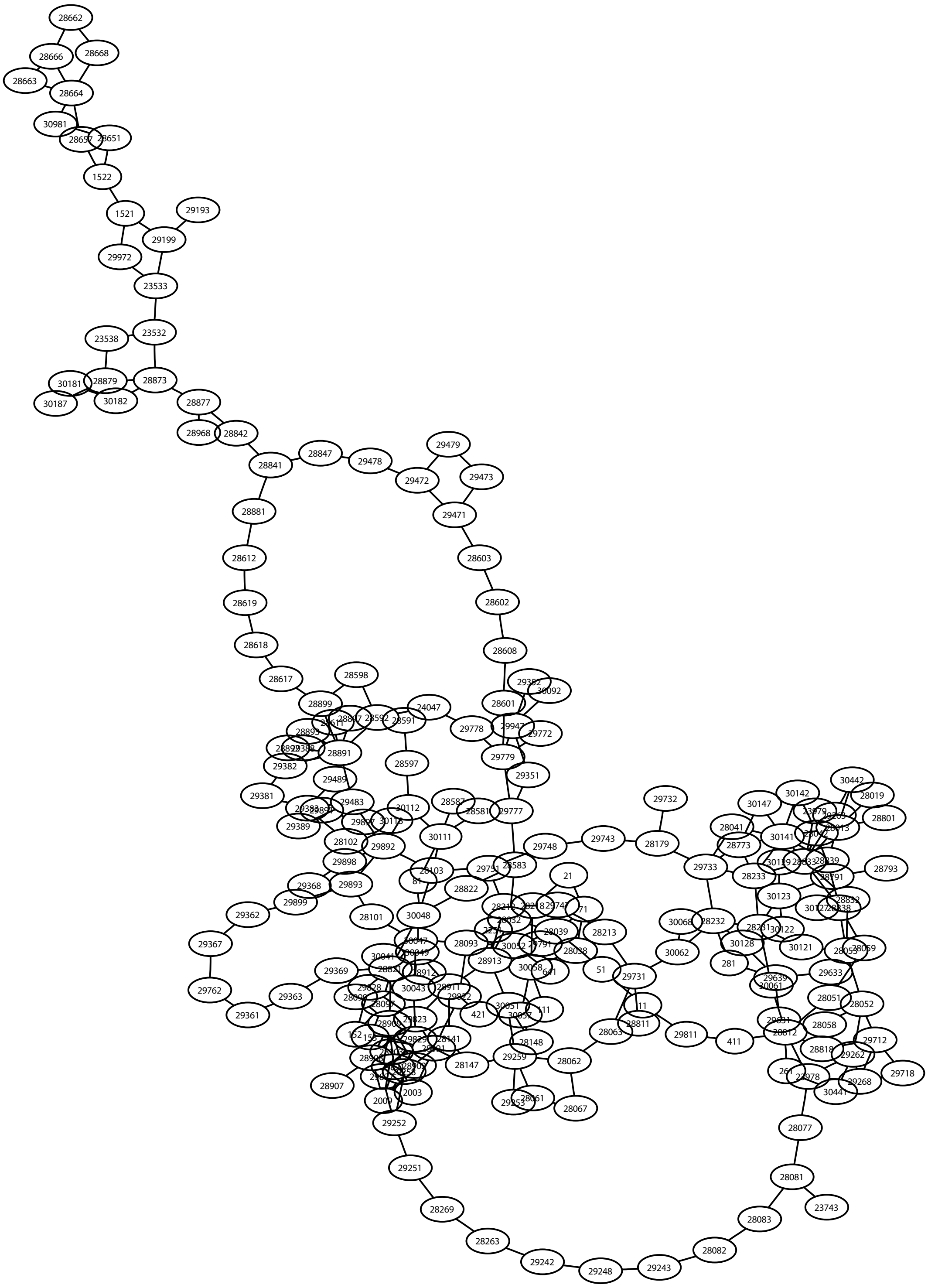


Figure 4.2: Logical network topology of section of Vodafone IT Rome GSM network, as extracted from recorded network traces

Network	Observed LA Size
Telstra Sydney	49
Telstra Melbourne	22
Orange London	15
Vodafone Rome	44
Orange Paris	23
Orange Hong Kong	31

Table 4.4: Average location area sizes observed in GSM network traces

unpredictable oscillatory behaviour within a given region, discussed further in the following section. From a highly practical viewpoint, this also indicates that cell transitions cannot be determined simply by a record of the change in cell name as displayed on a researcher's mobile phone [CLSS00], as many cell transitions may occur within a region sharing a common cell name. A more complex software package, such as that described in Section 4.1.1, is required to reliably record movement behaviour based on changes in cell *ID*.

The common occurrence of overlapping cells, coupled with the significant influence they have on movement characteristics, casts the relevance of traditional uniform network approximations into further doubt. The absence of such influential network parameters prevents such models from effectively simulating the characteristics of modern cellular networks.

4.1.4 Oscillatory Cell Behaviour

A user class deserving independent consideration is that of the non-mover - users who are either at work during the day or asleep at night, and are hence conducting no physical movement. According to current physical-based models, these users possess a movement velocity of zero, making no cell transitions. The modern network reality however is that stationary users make many cell regular transitions. This is the result of each physical location being covered by a great many cells, each with continually varying signal strength. Current mobile devices continually switch cells at idle to maximise available signal strength, resulting in such seemingly random cell transitions.

Date	Time	Cell ID	Location Area Code	Cell Name
1/08/2004	6:23:27	15622	8720	Parkville
1/08/2004	6:29:01	5360	8720	Melb CBD
1/08/2004	6:29:49	15622	8720	Parkville
1/08/2004	6:30:06	5632	8720	Carlton
1/08/2004	6:30:31	4493	8704	Brunswick
1/08/2004	6:30:38	5360	8720	Melb CBD

Table 4.5: Cell oscillations for a stationary user in a highly dense environment
Melbourne CBD, Telstra Melbourne GSM Network

Date	Time	Cell ID	Location Area Code	Cell Name
6/07/2004	6:09:09	7421	277	Kenthurst
6/07/2004	6:09:26	7423	277	Kenthurst
6/07/2004	6:10:22	7421	277	Kenthurst
6/07/2004	6:10:41	7423	277	Kenthurst
6/07/2004	6:24:30	24951	277	Kenthurst
6/07/2004	6:24:46	7421	277	Kenthurst

Table 4.6: Cell oscillations for a stationary user in a sparse rural environment
Rural Kenthurst, Telstra Sydney GSM Network

This transitional cell behaviour is not isolated purely to highly populous urban areas. Current network implementation trends collocate multiple cells at each physical transmitter, leading to devices in almost all areas of mobile coverage being able to swap between multiple cells in maximising signal strength. Tables 4.5 and 4.6 show samples of such cell ‘swapping’, in both urban and rural environments for a stationary user. Such patterns are present throughout all recorded network traces, with no lulls present where cell movement ceases entirely.

The source of cell oscillations is demonstrated in Table 4.7. Here the continually varying strength for a single cell is shown to vary significantly over a short period of time. This data was obtained using the CellTrack utility introduced in Section 4.1.1. The signal variance for even a completely stationary user points to a wide variety of unpredictable physical factors influencing signal interference, outside the realm of interpretation within a location management scheme.

Date	Time	Cell ID	Signal Strength
9/08/2004	12:04:06	23952	100%
9/08/2004	12:06:50	23952	73%
9/08/2004	12:08:23	23952	43%
9/08/2004	12:11:12	23952	66%
9/08/2004	12:12:53	23952	48%
9/08/2004	12:16:11	23952	24%

Table 4.7: Signal strength modulations for cell 23952, Newtown, in Telstra Sydney GSM Network

4.2 Evaluation of SUMATRA Traces

SUMATRA (Stanford University Mobile Activity TRAces) [PLE04] is a set of network traces produced to provide a common simulation "benchmark" on which to compare performance results of various mobile computing developments. While not offering recorded traces from real networks, these traces provide artificially generated user movement and call arrival patterns based upon a large body of research in user network behaviour. Long utilised within the network community to evaluate Location Management proposals, SUMATRA is purported to be unique in being closely validated against real data on calling and mobility traces [PLE04]. A dearth of available real-world network traces has resulted in artificial traces such as SUMATRA remaining the sole feasible mechanism for evaluating LM performance in a realistic network. Little research has been conducted however into the suitability of such artificial trace benchmarks.

The research behind SUMATRA [LJCW96] was conducted in 1995, an era when cellular communication networks were in their infancy, particularly in the United States where this research was conducted. By the authors' admission "very little [was] known about the traffic characteristics of future PCS networks," and hence call arrival probabilities were generated based on traditional wired telecommunication systems, with cell movements determined by sophisticated activity-based scenarios. The significant age of the SUMATRA research, coupled with a clear lack of network data at its inception, mandates an evaluation of these traces before their utilisation in a comprehensive cellular network simulation.

Parameter	BALI-2 Trace	Sydney Trace
Trace Duration (hours)	24	1250
Number of Cells	90	437
Number of Users	66550	6
Moves per Hour per User (mean)	0.15	12.4
Calls per Hour per User (mean)	0.98	variable

Table 4.8: Network parameters for SUMATRA BALI-2 trace and Telstra Sydney network trace

4.2.1 BALI-2 Trace Characteristics

The SUMATRA trace collection encompasses four distinct traces for modelling events in a mobile network:

- SULAWESI-1: Stanford U. Local Area Wireless Environment Signalling Information (small)
- SULAWESI-2: Stanford U. Local Area Wireless Environment Signalling Information (large)
- BALI-1: Bay Area Location Information (peak time)
- BALI-2: Bay Area Location Information (real-time)

The most relevant and comprehensive trace is the BALI-2, a 24 hour call and movement event trace for the San Francisco Bay Area cellular network. The aggregate network characteristics for this trace are listed in Table 4.8, along with the corresponding parameters recorded from the survey of the Telstra Sydney GSM network. Note that the call rate parameters for the Sydney trace are absent, owing to their significant variability per-user and hence inability to be accurately recorded with such a small user base.

Immediately apparent is the large differential in the mean movement rates for users - the mean time between individual user movements equating to 4.8 minutes in the Sydney trace and 6.6 hours in SUMATRA, a factor of over 80 times longer than that seen in the Sydney network. When compared to denser networks, such as the Orange GSM network in London, with an average cell movement rate of 28.6 moves/hour, this factor rises to over 190, signalling a highly unrepresentative cell movement rate.

Without the resources to reliably record a wide range of call arrival rates in a real network, it is not possible to conclusively evaluate the appropriateness of the call rates implemented within SUMATRA. These call rates were based on analysis of call frequencies in wired telephone network, along with call distribution per time-of-day and user class. While an average of 24 calls per day for a typical user in a cellular network seems optimistic, these assumptions cannot be combated here with hard evidence. It is certainly feasible that with continued growth and lowered cost of cellular networks user call rates will increase to such a level and beyond.

The individual user movements within SUMATRA are generated according to complex activity-based models. This formulation results in a well varied set of movement data. The activity-based model also produces directed periods of user movement, contributing to concrete network movement probabilities. While this high variance is suited well to the task of modelling accurate user movement, the coupling of physical- and network-based movement detracts from its true representative capabilities. While 'ping-ponging' cell oscillations are visible for all users in the real GSM traces of Section 4.1, no such network-centric characteristics are present within the SUMATRA BALI-2 trace. Coupled with the extremely low cell movement rate, little real correlation may be seen between these artificial traces and real GSM network data.

4.2.2 SUMATRA Weaknesses

The shortcomings in SUMATRA, as demonstrated by a highly uncharacteristic user movement rate, are a manifestation of antiquated modelling assumptions made extensively throughout the LM research community. Such assumptions tightly couple the relation between the physical movement of users and their movement between cells in a cellular network. This offers a convenient reference frame for prediction and parameterisation of user movements, yet does not apply well to modern cellular networks replete with dense overlapping cells. While previous research has shown that the direction of cellular movement is not directly related to user activity [CLSS00], the data presented in Section 4.1 conclusively demonstrates that there is no one-to-one correspondence between physical movement velocity and cell crossing rate.

[Kol03] has found the mean LA size for an ideal static location area mapping on the SUMATRA BALI-2 trace to be in fact a single cell. Such a characteristic precludes the very existence of dynamic location management techniques, with a single cell LA system equating to a network with no location management at all. Clearly this is a highly undesirable characteristic of trace data used to evaluate the very performance of location management schemes, and once again related to the low cell movement rate. When compared to a mean static LA size observed in the Sydney Telstra GSM network of over 72 cells, the inaccuracy of such network characteristics is highlighted.

The general movement trends in SUMATRA are well formulated however, providing user movement description beyond the realm of simple artificial trace generation. Call distribution is also based on in-depth research into wired network usage patterns, obeying time-dependent characteristics rather than simple Poisson distributed events. For these reasons SUMATRA remains a valuable source of simulation data. A modification of movement rates and patterns to reflect a network-centric view of user movement, replete with spontaneous hopping between neighbouring cells, would provide a truly solid location management benchmark.

4.3 Alternate Trace Generators

With the commonly used SUMATRA BALI traces exhibiting characteristics far from representative of modern cellular networks, a requirement remains for trace data to supplement SUMATRA. With no realistic trace generators available, as mentioned previously, this task is greatly complicated. Two approaches to providing useful generation of user activity are discussed below, used in the evaluation of location management results in the following chapter.

4.3.1 Physical Movement Models - MobileManager

The majority of research into location management is still entrenched in physical movement models, characterising user movement through geographical coordinated and movement vectors. This research direction is reflected in available benchmarks; these using physical user models to generate trace data. While much of this research is limited in its approach, one highly flexible package for generating semi-realistic user data is the recently developed MobileManager trace generator [TZss].

MobileManager is itself based on the same research used in the development of SUMATRA [LJCW96] and hence shares many of its less ideal characteristics of low movement rate and lack of random 'frequency-hopping' cell transitions. While SUMATRA comprises a series of static traces however, the MobileManager system dynamically generates traces based on a large number of user inputs, allowing output to be customised towards a more representative set of trace data. In particular a modified version of MobileManager, allowing direct modification of user velocity, is used in the simulations of Section 5.4. Here mean user velocity is increased from the default of 20 units to 200, allowing a higher user movement rate simulating a modern dense network.

A huge variety of user characteristics are available for use here, with the default values based on previous research into user behaviour and used in the majority of cases. A significant advantage for the testing of heterogeneous user distributions is the provision of three discrete user classes: *worker*, *ordinary* and *commuter*, with decreasing regularity in movement patterns.

Arbitrary network topologies are able to be constructed through the generator, with both traditional hexagonal grid and a more realistic heterogeneous topology considered in the evaluation of location management performance. The construction of these topologies is discussed in more detail in Section 5.4, used to evaluate the proposed scheme against both traditional idealistic and modern networks.

While a large degree of control over simulation parameters is available, little direct control over mean movement and call arrival rates is possible. In compensation, the trace output modified before use during the simulations of the following chapter. Here trace duration is compressed in time to allow for variation in movement rate, while calls are selectively dropped to control mean call arrival frequency. This manual modification of the complex trace data detracts from the temporal movement characteristics exhibited, such as users travelling to and from work at specified times, yet remains the only feasible option for efficiently generating a large trace data set covering a wide scale of call and movement parameters.

4.3.2 Random Parameterisation

While trace generators provide realistic user characteristics over a small range of movement and call arrival rates, they do not allow the wide range of parameterisation required in examining the performance of an abstract location management scheme. The proposed location management scheme is claimed to function ideally for any cellular network formulation, a claim poorly supported via testing over a small range of network and user parameters.

The use of a purely random trace generator allows the efficient production of a large set of user data, conforming to greatly varied system parameters. While sophistication of user movement is not present in this simple 'random walk' behaviour, the simple formulation of user data allows an investigation of location management stability under a range of network conditions. The desire to apply the proposed LM scheme to future Mobile IP networks as discussed in Chapter 7 requires a scheme capable of dynamically accommodating for such varied user parameters.

Section 5.2 will examine location management performance over such a random parameterisation scheme. Here call rate may be varied from 0.025 to 12.8 calls/hour with movement rate a variable 0.2 to 102.4 moves/hour. While a random movement benchmark is certainly not ideal when relating performance to a real-world network, only such a formulation is able to provide the desired flexibility in performance analysis

4.4 Summary

This chapter has presented a number of simulation options available for benchmarking location management performance, evaluating their suitability before implementation in the following chapter. The most interesting result from this evaluation however is the large differential in user movement rates between traditional network models and real cellular networks, as recorded in GSM network traces. This large difference is primarily owing to the higher density of currently implemented networks, and propensity for mobile devices to 'hop' between overlapping cells in maximising signal strength. The poor correlation discovered between physical movement and network cell-crossings raises doubts to the suitability of traditional network models, necessitating the use of a highly varied set of user parameters when evaluating specific location management performance.

Chapter 5

Simulation and Analysis

This chapter will investigate the performance of the proposed location management scheme under a variety of disparate network environments. Performance evaluation is conducted through the use of a simulation engine for cellular networks, developed for evaluating location management proposals and discussed in Section 5.1. These simulations are run on both real network traces and artificially generated data, used to evaluate the comparative performance of three major location management schemes:

- **Per-User Dynamic**

  The full dynamic location management scheme, as proposed in Chapter 3. This scheme considers individual user call rates and movement factors, along with aggregate movement probabilities at each cell, to determine an optimal allocation of location areas based on both user and network. The per-user dynamic scheme is proffered as the most suitable for implementation in modern cellular networks, offering the greatest possible reduction in location management cost.

- **Aggregate Dynamic**

  An idealised dynamic location management scheme considering only aggregate characteristics, as opposed to individual per-user behaviour. While this scheme is able to benefit from the automatic discovery of cell neighbours, along with the learning of dwell times and call rates, these characteristics are considered only as an average across all users. Movement probabilities between cells are hence considered equal and all users are assumed to have identical parameters. This scheme represents a perfectly configured dynamic scheme sans per-user characterisation, used to allow evaluation of the specific impact of per-user optimisation on top of an otherwise ideal dynamic location management scheme.

- **Static**

  Location areas are predetermined in a static overlay in this scheme, representing location management in current GSM network implementations. While the location area size is fixed amongst a set of simulations, this size is determined as the minimal cost static LA assignment for the most commonly expected user characteristics. Unless specified to the contrary these parameters comprise a movement rate of 20 moves/hour and a call arrival rate of 0.5 calls/hour per user, values based on the research conducted in Chapter 4.

In all cases the communication of a dynamic location area to a user device is considered *perfect* and *free*. It is assumed that no errors occur in the representation and transmission of each LA, and that no communication cost is involved. This is to simplify analysis and to present results in line with current location management research. The specific impact of location area communication is considered explicitly in Chapter 6.

95% confidence intervals are shown for all results, with the exception of deterministic simulations such as the pre-recorded move and call traces, where there is no variance upon which to judge deviation in successive iterations.

5.1 Simulation Framework

A comprehensive discrete-event simulator for cellular networks was developed and used to evaluate the performance of the various location management schemes. This simulation engine is highly modular, allowing for customised *generators* and *LA specifiers*, enabling the evaluation of network and user specifications under a number of location management schemes.

Each *generator* defines a mechanism for registering call and movement events with the network. A generator may parse pre-existing trace data from a file, or automatically generate trace data based on variable parameters. The generators implemented in the simulation engine are defined as:

- **TraceGenerator**

  Generator reading entire artificially generated trace contents from file, including full movement and call events - used with all SUMATRA-based traces [PLE04].

- **ModTraceGenerator**

  Modified TraceGenerator, selectively dropping call events and inserting sporadic 'ping-pong' cell movements to customise mean call arrival and movement rates from SUMATRA data while retaining general user patterns.

- **AutoGenerator**

  Automated random user movement, negative exponentially distributed around given call arrival and movement rates, performed over SUMATRA BALI topology.

- **RealTraceGenerator**

  User movement events as recorded from network traces of real GSM networks, seen in Section 4.1, with network topology generated directly from these traces. Call arrivals are negative exponentially distributed around a given value.

- **StaticAutoGenerator**

  AutoGenerator functionality over SUMATRA BALI network with optimal static location area overlay. This is used as a customisable comparison to currently implemented networks.

- **RealStaticTraceGenerator**

  RealTraceGenerator with an overlaid static location area scheme as recorded directly from each surveyed network. The use of this generator allows a record of the exact cost imparted on a network by recorded cell movements.

Acting in conjunction with these generators are a number of *LA specifiers*, defining the algorithm used when determining the location area assigned to each user. The specifiers used in development and performance evaluation are:

- **ProbLA**

  Full per-user dynamic location management scheme implementation, as proposed in Chapter 3. Each location area is assigned dependent on aggregate movement probabilities at each cell, along with the specific call rate and movement factor of each user.

- **AdjLA**

  Aggregate dynamic location management implementation, representing the optimum LM scheme available without considering individual user behaviour. Location areas are assigned based on average parameters amongst all users, with random movement probabilities between cells.

- **StaticLA**

  Static location management implementation. When used with StaticAutoGenerator or RealStaticGenerator this scheme will assign predefined location areas as recorded in trace data. Under all other generators a location area will approximate a circular region, of fixed size specified as a system parameter.

- **SingleLA**

  Simplistic location area scheme assigning a single cell to each user. This simulates a network with no location management system in place.

- **AllLA**

  Simplistic location area scheme assigning the entire network topology as a location area to each user. This simulates other extreme of location management approach to that presented by SingleLA.

- **ProbBloomLA**

  Per-user dynamic location area implementation with the false-positive compensation mechanisms of Section 6.4 incorporated to evaluate the performance of a Bloom filter LA communication system.

A number of associated programs were developed to automate the process of formatting trace data, analysing network parameters from recorded traces, and performing multiple varied simulations along with 95% confidence intervals. The results output from this simulation package, along with their analysis in the context of location management performance, are presented below.

5.2 Performance for Random User Movement

While trace data containing realistic user behaviour is of prime interest, it remains important to examine the performance of the proposed location management system for completely random user movement. Such a random formulation relaxes all assumptions on regularity of user patterns and allows a concrete evaluation of performance in an 'unpredictable' cellular network. Interesting from more than a purely theoretical perspective, [CLSS00] shows that such seemingly random movement is a common characteristic of dense urban networks. This examination of random user movement will form the initial basis of evaluation for location management performance.

The network topology used in these simulations is that taken from the SUMATRA BALI traces, representing a realistic complex network structure. Mean call arrival and movement rates are varied over each iteration, with individual user characteristics deviating from these averages via a Poisson distribution. These simulations aim to exhibit the ability of the proposed scheme to conform to network structure and user rates, without yet considering the complexity of individual user movements and aggregate movement patterns.

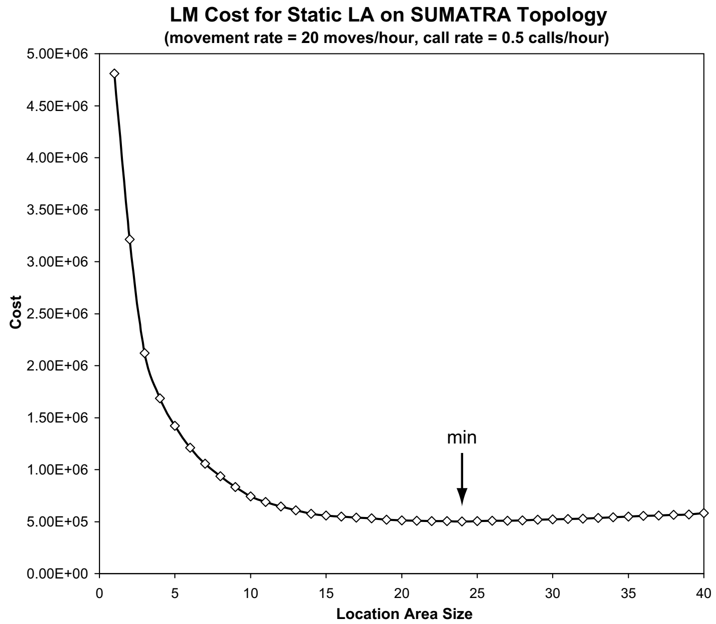


Figure 5.1: LM cost for randomised movement over SUMATRA BALI topology with fixed LA size

5.2.1 Determining Static LA Overlay

Before the performance of a dynamic scheme may be compared to a traditional static location area system, an ideal static location area overlay must be determined for our given topology. Here no existing static scheme exists for the SUMATRA topology, and hence one must be manually generated. An ideal static location area need minimise the cost for an average user, assumed here as 20 moves/hour and 0.5 calls/hour. Note that mean user characteristics are varied across a wide variety of parameters in the performance comparisons, and hence the static scheme will approach the ideal approximation for only one set of simulation values, regardless of the assumed average rates.

Figure 5.1 shows location management cost for the assumed average user parameters, with variable fixed location area size. It can be seen here that the minimum network cost occurs for a location area size of 24. The shallow tail exhibited beyond this value indicates a relatively small cost penalty for sizes larger than this value - a larger LA leading to less frequent *costly* location updates and only a minimal increase in relatively *inexpensive* paging overhead.

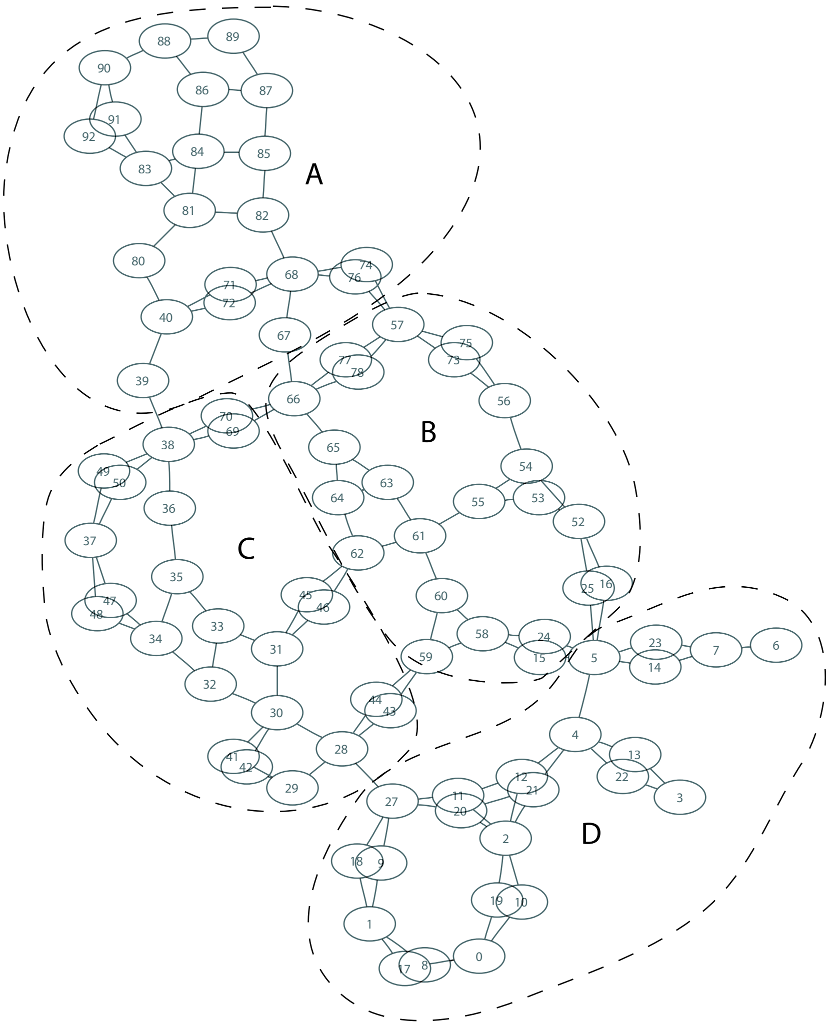


Figure 5.2: Static LA scheme overlaid on SUMATRA BALI topology

This ideal fixed location area size is used to generate a static location area overlay on the SUMATRA BALI topology, demonstrated in Figure 5.2. Here the network is partitioned into 4 equally sized static location areas of 23 cells each, allowing an even partitioning of cells with location area sizes very close to the ideal value.

5.2.2 Varied Call Arrival and Movement Rates

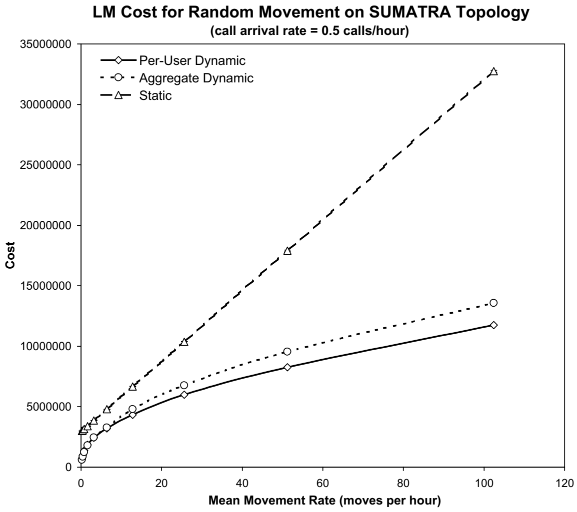
The performance of the per-user dynamic, aggregate dynamic and static location management schemes for random user movement are examined in Figures 5.3 and 5.4. The simulations of Figure 5.3 vary mean user movement rate between 0.2 and 102.4 moves/hour with call rate fixed at 0.5 calls/hour. The linear-scale plot of Figure 5.3(a) clearly shows a linear increase in cost for the static scheme while cost increase is much reduced with increasing movement rate for both dynamic schemes.

Despite a lack of user movement patterns, the per-user dynamic LM scheme is able to consistently reduce cost over an aggregate dynamic scheme, for mid to high movement rates. This advantage is provided by the per-user scheme learning the call and movement characteristics of individual users, as distributed around the specified means, and assigning location area sizes accordingly. This reduction in cost is only moderate owing to a lack of user movement patterns and little variance in user characteristics.

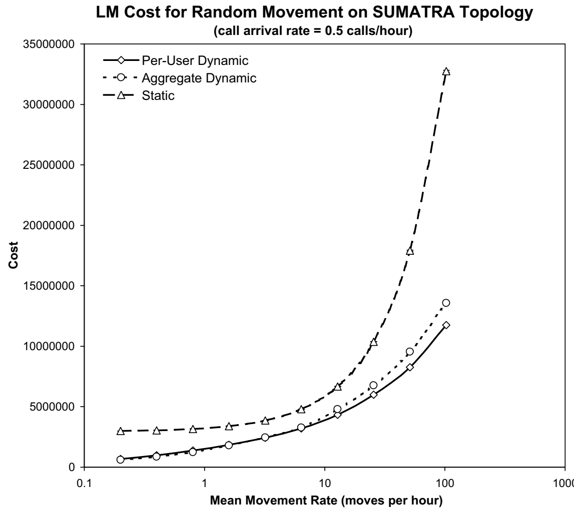
The log-scale plot of Figure 5.3(b) shows the convergence of the per-user and aggregate dynamic schemes at low movement rates. At extremely low movement rates of less than one cell crossing per hour, the location update rate is so small so as to not convey enough information to the network to build an accurate profile of user characteristics. A result of this poor granularity is the degradation to an aggregate dynamic scheme where little difference between individual users is recognised. For movement rates expected in modern networks, with over 10 moves/hour, a clear performance differential is present.

Also visible from the log-scale plot is that the performance of the static location management scheme most closely approaches the performance of the dynamic schemes when average movement rate is around 10-20 moves/hour. This is an intuitive result reflected in the static location area overlay optimised for 20 moves/hour. When movement rate deviates from that assumed in the development of the static overlay, performance is seen to suffer dramatically.

Figure 5.4 shows location management cost in both linear and log-plot scales for varied mean call arrival rates between 0.025 and 12.8 calls/hour, with a mean movement rate of 20 moves/hour. The results here are very similar to those presented in Figure 5.3, supporting the inferences made thus far. Note that while cost increases linearly for a static scheme where movement rate is varied, the cost presented here is seen to increase sub-linearly for high call-arrival rates. This is owing to the comparatively small impact of the call-rate dependent paging frequency on network cost, in comparison to movement rate which affects location



(a) Linear plot



(b) Log plot

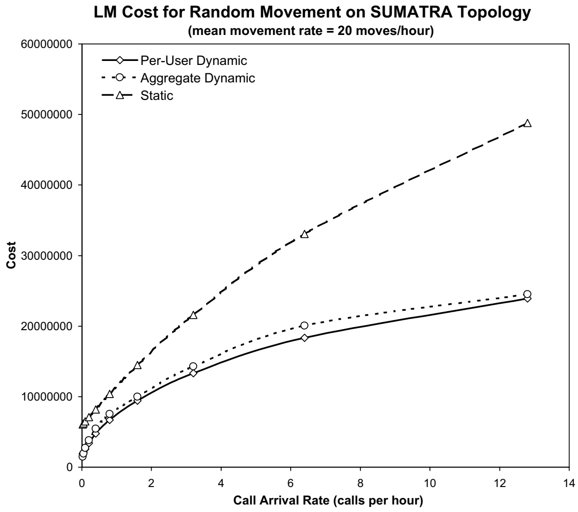
Figure 5.3: LM cost for random movement over SUMATRA BALI topology with an average 0.5 calls/hour

update frequency and is seen to dominate the network cost function.

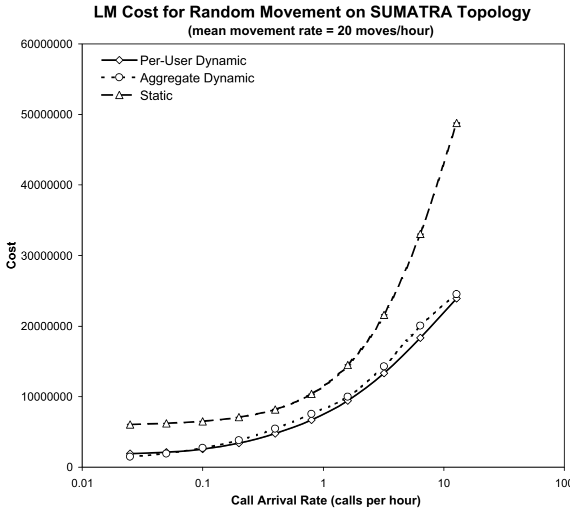
5.3 Performance with Traditional Metrics

Although traditional network benchmarks have been seen to relate poorly to modern cellular networks, as demonstrated in Section 4.2, it is nonetheless important to consider their performance. It is claimed that such benchmark traces are unrepresentative of real networks, yet they remain the only point of comparison between the proposed scheme and traditional location management systems. Traditional location management benchmarks, despite their poor parameterisation, also contain a depth and complexity of user movement difficult to capture from artificial traces.

These simulations will examine the performance of the static and dynamic location management schemes for the commonly used SUMATRA BALI-2 cellular network trace [PLE04]. This record of call and movement events over a 24 hour period in a large network of 66550 users is examined in depth in the previous chapter.



(a) Linear plot



(b) Log plot

Figure 5.4: LM cost for random movement over SUMATRA BALI topology with an average 20 moves/hour

5.3.1 Determining Static LA Overlay

As in Section 5.2.1, an ideal static location area assignation must be determined before the performance of traditional static metrics may be compared to proposed dynamic systems. Figure 5.5 shows the variance in cost for fixed-size location area allocations in the SUMATRA BALI-2 traces. Note here that the ideal static location area size is in fact a single cell, a consequence of the unrealistically low SUMATRA movement rates examined in Section 4.2. This surprising result is supported directly by the work of [Kol03], and hence a single cell static scheme will be used when simulating a static location area mechanism for the BALI-2 traces.

5.3.2 SUMATRA BALI-2 Benchmark

Since BALI-2 is a complete deterministic trace of call and movement data, no variation of these parameters is possible in running multiple simulations. Instead the ratio between individual location update and paging message cost $\varsigma_{LU} : \varsigma_P$ is varied across each simulation. This ratio,

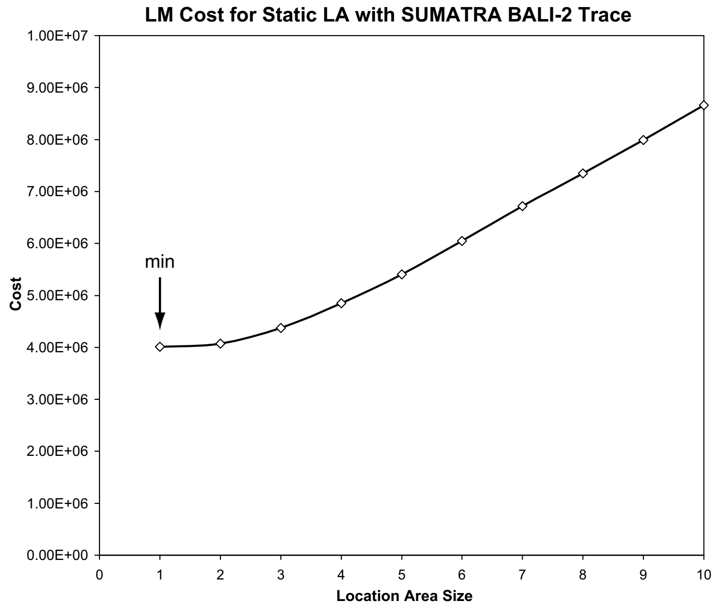


Figure 5.5: LM cost for SUMATRA BALI-2 trace with fixed LA size

discussed in Section 3.7, is merely *estimated* at 10:1, and not well understood for modern networks. By varying this ratio across the simulations, an indication of performance across a wide variety of network implementations is possible.

Figure 5.6 shows the performance of the aforementioned schemes on the SUMATRA BALI-2 trace. As expected, performance is indistinguishable between the three schemes for $\varsigma_{LU} : \varsigma_P$ ratios near the accepted 10:1. Since the ideal location area in this case is at most a single cell, there is little room for improvements made by the dynamic schemes. As this ratio increases however, the comparative cost of paging decreases, raising the average size of each location area above a single cell. In this new scenario there exists scope for the dynamic schemes in reducing location management cost, as exhibited by the divergence of the cost plots.

Regardless of the cost ratio between location updates and paging, the mean movement rate in all simulations remains the 0.15 moves/hour implicitly defined by the trace data. This low granularity of per-user movement information prevents the per-user dynamic scheme from exhibiting large performance improvements over an aggregate dynamic scheme. With such a low movement rate the average user makes a mere 3.6 moves during the 24 hour period of the trace, providing little opportunity for learning user characteristics and using these during

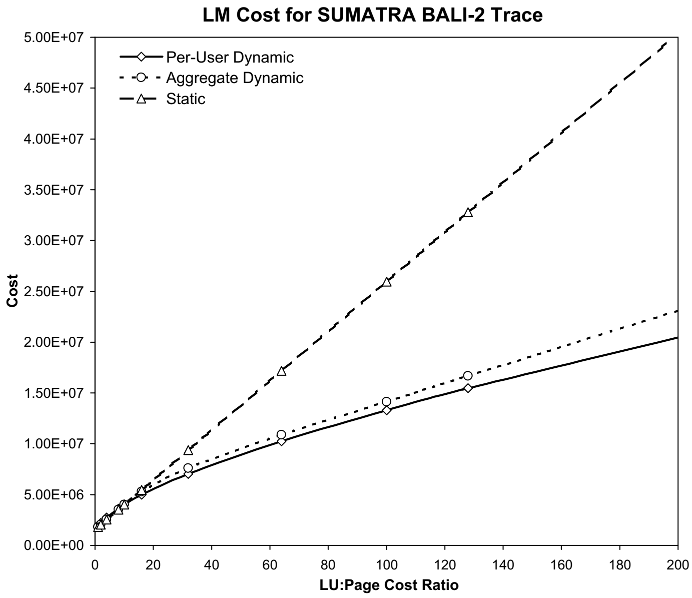


Figure 5.6: LM cost for SUMATRA BALI-2 trace with static and dynamic schemes

possible location updates made on subsequent moves.

Nonetheless, a clearly discernable reduction in cost is proffered by the per-user dynamic scheme over the aggregate formulation, exhibiting the per-user scheme's stability in the presence of very low granularity user data. The GSM network survey of Section 4.1 identifies an average of 297.6 moves per user in a similar 24 hour period for the Sydney Telstra GSM network, with even higher rates observed in other cities. The ability of the scheme to reduce LM cost for individual users based on an average of only one or two moves before a location update, signals great potential for use in current networks.

5.4 Performance with Modern Metrics

While the performance advantage of a dynamic location management scheme over traditional static systems has been shown conclusively thus far, what remains is to determine the comparative performance of aggregate dynamic and per-user dynamic location management. Traditional models do not possess the frequency of movement and variation in user characteristics required to suitably compare per-user and aggregate dynamic schemes, necessitating the use of more representative benchmarks.

5.4.1 Simulation Parameters

While no trace generators reflecting the characteristics of modern GSM networks currently exist, recent generator developments may be modified to approach a realistic set of user data. The MobileManager trace generator [TZss] presented in Section 4.3.1 allows the customised generation of complex cellular network trace files, based on a wide range of user parameters and topology descriptions. While, in essence, producing data based on the research used in the development of SUMATRA [LJCW96], and hence reflecting the same unrealistic cell movement rates, the output from this generator may be modified to closely approximate that which may be expected in a realistic cellular network.

Two trace environments are investigated here, one representing a traditional 'honeycomb-like' homogeneous topology and the other a more typical heterogeneous network topology, both populated with realistic user movements. The generated populations of around 700 users produced by MobileManager are modified to possess a mean velocity of 200 'units', resulting in a per-user mean movement rate of 1.35 moves/hour and call arrival rate of 0.23 calls/hour over an 8 day period. These traces are compressed in time by a factor of 8 to result in 24 hour traces with a more realistic 10.8 moves/hour and 1.85 calls/hour, while retaining variation between individual users and complexity in user movement. Calls are then selectively dropped in each of the traces, allowing variation in mean call rate while maintaining temporal call distributions.

5.4.2 Homogeneous Network Topology

The homogenous network topology used to generate realistic movement on traditional network models is shown in Figure 5.7. This model is used to demonstrate the performance of a per-user dynamic scheme where no pronounced heterogeneity in cell characteristics is present. The model also allows comparison with traditional simulations restricted to such simple topologies,

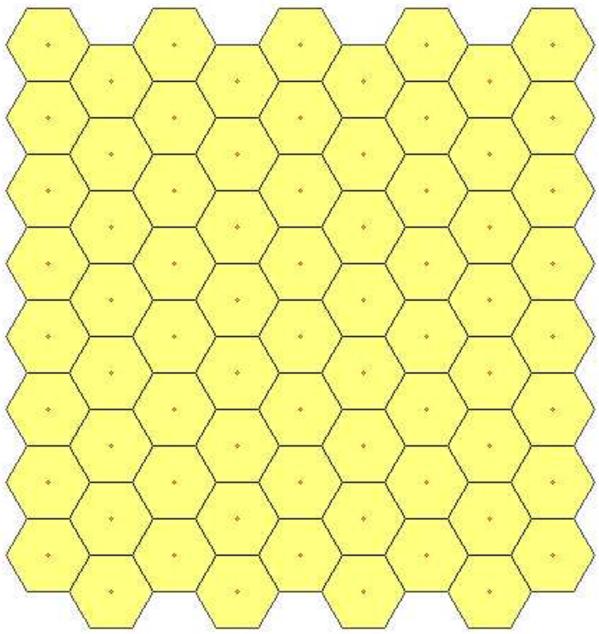


Figure 5.7: Homogeneous network topology used to generate semi-realistic artificial user movement

demonstrating that the cost benefit of per-user dynamic location management still holds under existing models used within the research community. User residences are located toward the top of this topology, while workplaces are situated toward the bottom, to ensure a significant and realistic degree of user movement for users belonging to the 'worker' class produced by the MobileManager trace generator.

Figure 5.8 demonstrates the performance for both aggregate and per-user dynamic schemes over this regularly distributed topology. Here the performance gain provided by the per-user dynamic scheme over a purely aggregate system is far more pronounced than in the previous simulations. This is owing to a far greater range and complexity of user behaviour, able to be exploited by the per-user dynamic scheme. As opposed to the experiments with random user movement, here clear aggregate movement patterns are present, such as workers travelling to and from their workplace, reflecting real-world networks and allowing further cost reductions. Note additionally that the consistency between simulations is much greater for the per-user

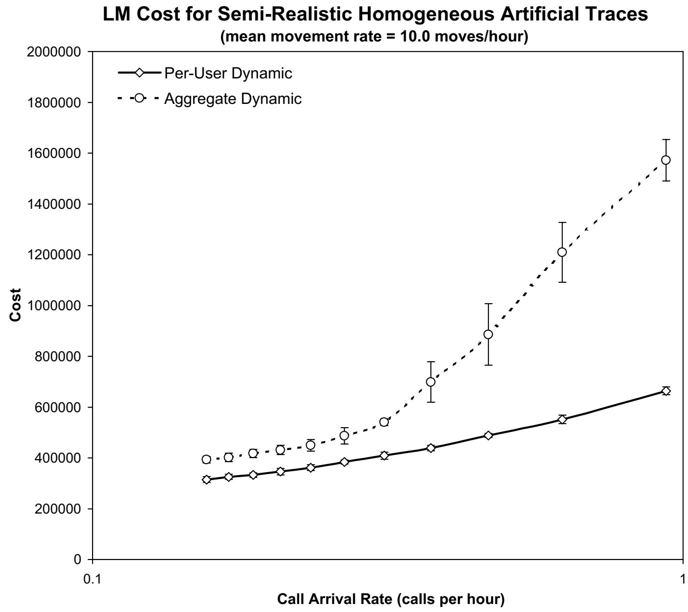


Figure 5.8: LM cost for semi-realistic artificial user movement on homogeneous network topology

scheme, able to compensate for differences in user characteristics, while the aggregate scheme is strongly affected by the degree in which users deviate from the mean parameterisation.

5.4.3 Heterogeneous Network Topology

A more realistic network topology is shown in Figure 5.7. This network map aims to approximate the complexity of modern cellular networks, with both dense and sparse network regions, overlapping cells, and irregular network boundaries. User residences are defined in MobileManager to be biased towards the sparse network sections, representing suburban and rural areas, while workplaces are located in the dense 'CBD-type' regions. Simulations performed over this topology aim to emulate a realistic network trace, with the flexibility of modifying call rates to examine performance changes as per-user network utilisation increases.

Here performance is very similar to that over the regular network topology, as presented in Figure 5.10. While the per-user dynamic scheme is able to function equally well for homogeneous and heterogeneous network topologies, demonstrating its ability to compensate for variations in network-based movement characteristics, the performance of the aggregate

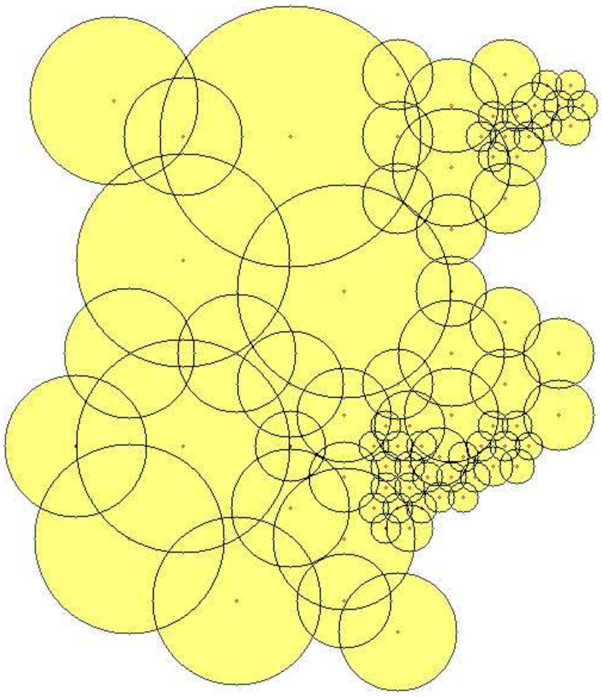


Figure 5.9: Heterogeneous network topology used to generate realistic artificial user movement

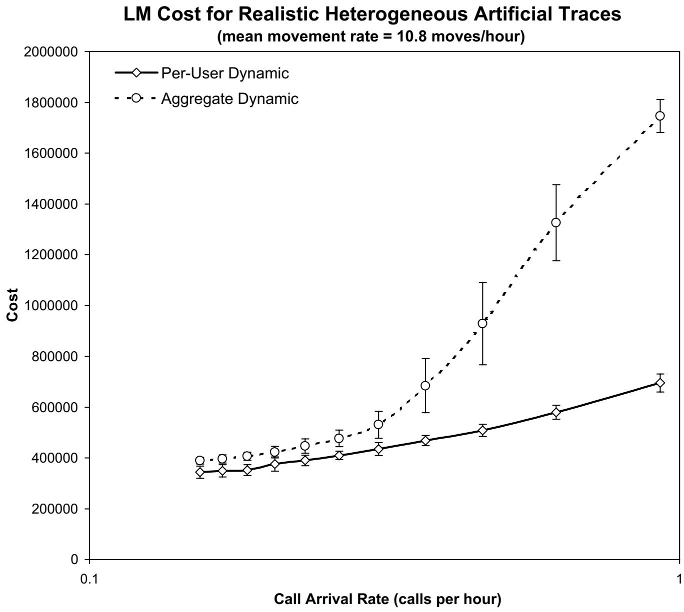


Figure 5.10: LM cost for realistic artificial user movement on heterogeneous network topology

scheme is somewhat reduced. The larger variation in cell crossings caused by the less uniform topology prevents the aggregate scheme from assigning appropriately sized and shaped location areas, causing a further increase in location management cost.

5.5 Performance in Real Networks

What remains is to evaluate the performance of the proposed location management scheme in a real cellular network. While the location management algorithm is abstracted to provide idealised performance in any type of cellular network, it is of no more than simple theoretical interest if this performance is not justified in the context of a realistic and typical cellular network. This evaluation is performed via simulation over the GSM network traces recorded from a number of mobile communication networks, presented in Section 4.1.

5.5.1 Typical Network Performance

While movement rates are fixed as recorded in the traces, calls are automatically generated in all simulation runs. This allows for investigation into performance against a variety of user call characteristics, while avoiding the poor granularity in recording trace call data with a very small user set. These calls are generated with negative exponentially distributed interarrival times, with mean rates ranging from 0.025 to 51.2 calls/hour.

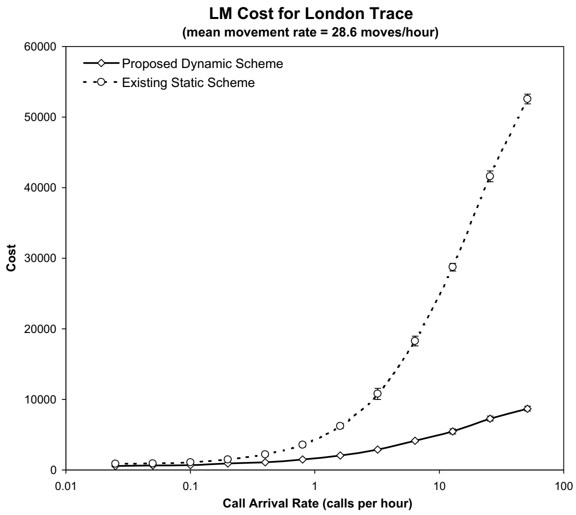
In these simulations both static and dynamic location management schemes are examined. The static scheme is extracted directly from the GSM trace data, with a change in recorded location area signalling a location update, and hence reflecting the exact location management cost imparted on the real-world network during these traces. Only a per-user dynamic location management scheme is considered, with no comparison made with aggregate dynamic performance. As the majority of traces record the movement of a *single* user, albeit over an extended period of time, the aggregate case matches the per-user case in almost all aspects of simulation.

Any disproportionate advantage the dynamic scheme gains from considering only a single user is mitigated by a lack of data provided in learning cell movement probabilities. This may be noted through the Sydney network simulations, presented in Figure 5.12, as run over 25 concurrent users with no appreciable variation in results. The exact number of users in each simulation may be observed as the number of sub-traces in the network trace descriptions presented in Appendix B.

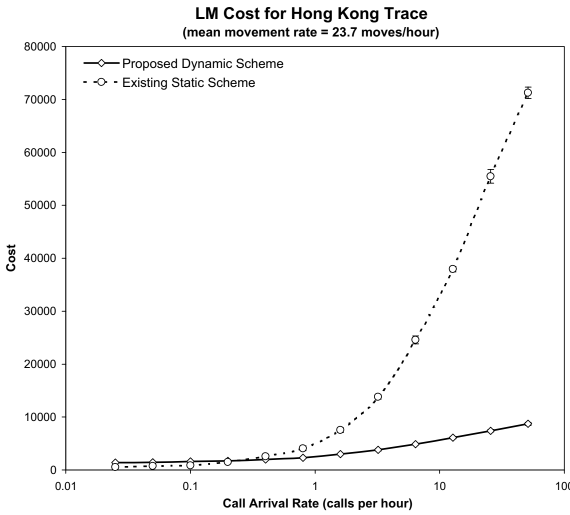
The results of the simulations over real network traces are visible in Figures 5.11 and 5.12. These plots reflect those shown previously, with the performance differential between static and dynamic schemes growing markedly with an increase in call rate.

5.5.2 Low Granularity Exceptions

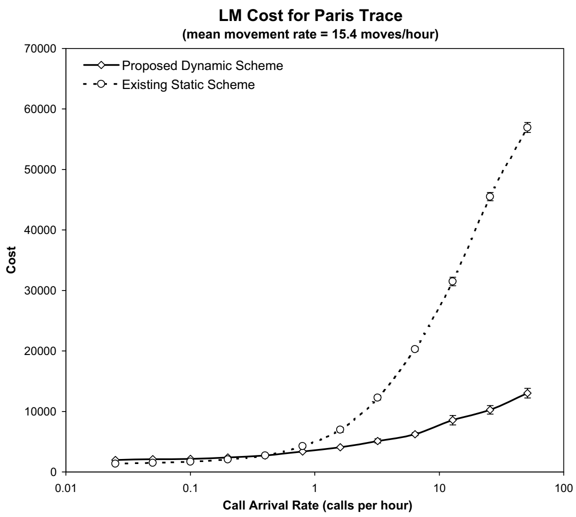
It should be noted that for the Hong Kong, Paris and Rome traces of Figure 5.11, performance is slightly worse with a dynamic location management system for call rates below 0.1



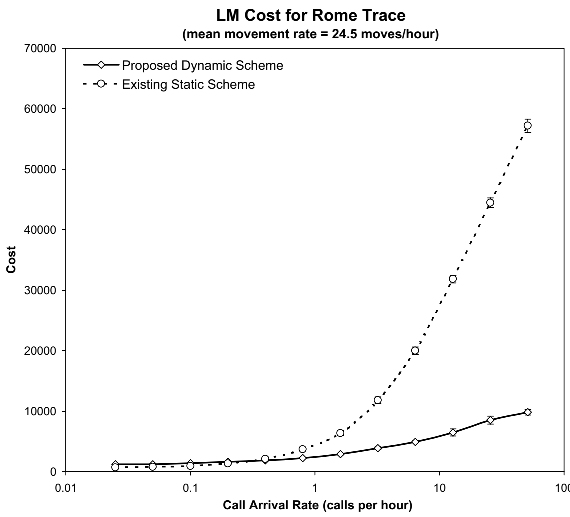
(a) Orange GSM Network, London



(b) Orange GSM Network, Hong Kong

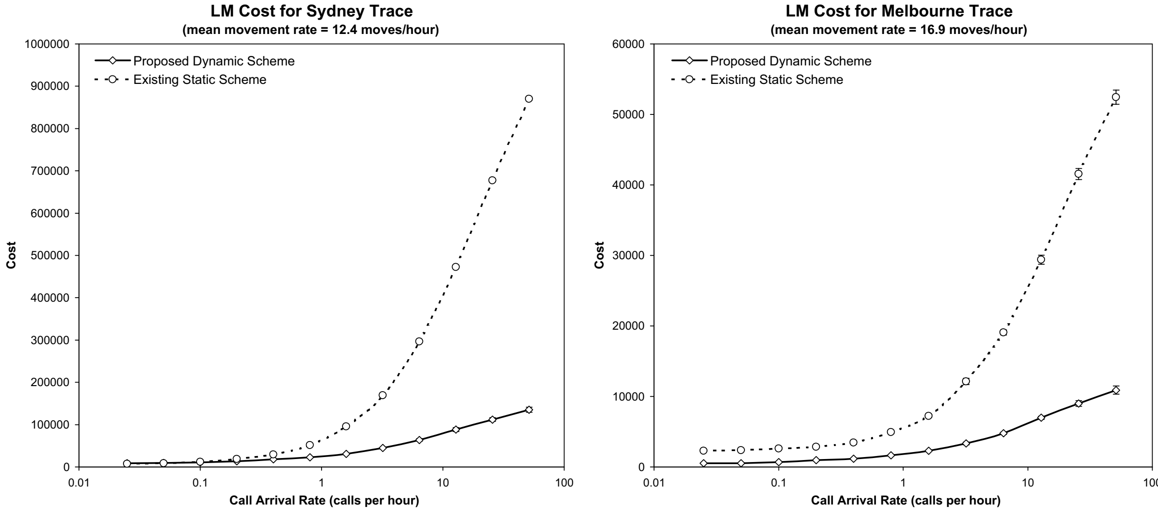


(c) Orange GSM Network, Paris



(d) Vodafone IT GSM Network, Rome

Figure 5.11: Performance comparison with existing static LM schemes (International)



(a) Telstra GSM Network, Sydney

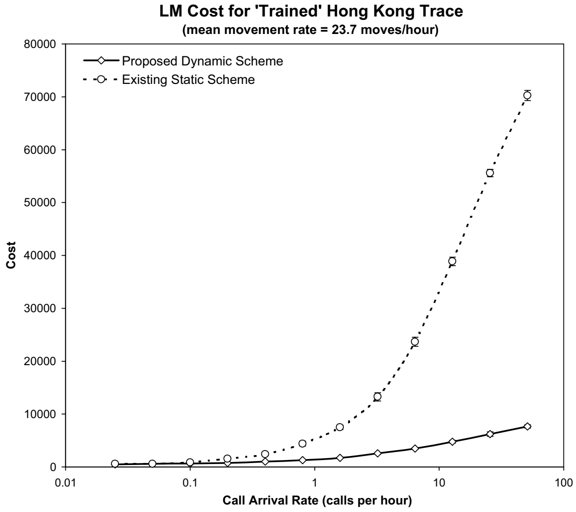
(b) Telstra GSM Network, Melbourne

Figure 5.12: Performance comparison with existing static LM schemes (Domestic)

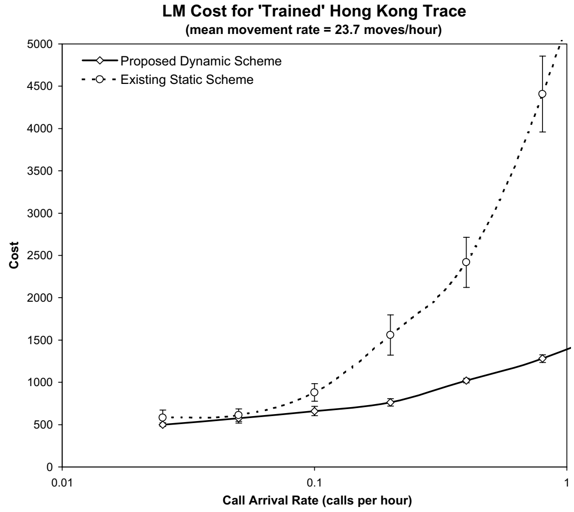
calls/hour. This effect is a consequence of the network not adequately learning its topology during the extremely short on-call proportion of the simulation.

In all simulations the network is initiated with cells having no knowledge of their neighbours, forced to gather this information dynamically. It is only during handoff however that a cell may learn of a neighbour, this handoff by definition only occurring when a user is moving between cells while on a call. The small trace duration, coupled with just a single low-call-rate user residing in the network, results in the majority of cells not given the opportunity to learn of the presence of their neighbours. Without at least a primitive knowledge of network topology, the search for cells to insert in a location area is truncated, restricting the ability to assign an adequate LA to the user.

In light of this information, the claim that dynamic performance degrades to an ideal static scheme should be amended, to include the caveat that each cell needs sufficient information to learn of its primary neighbours. Such a caveat is expected to never be observed in a real implementation however, with a larger number of users, longer trace duration, or more frequent call rate readily providing the requisite information to each cell. The satisfaction



(a) Regular scale



(b) Magnified scale

Figure 5.13: LM cost for Hong Kong GSM traces, allowing network *training* before simulation

of these criteria in the previous simulations shows the cost for a dynamic LM scheme never rising above that of an ideal static implementation.

In a theoretical, and highly unrealistic, system where none of these conditions are met, the problem of reduced dynamic LM performance may be avoided simply by informing each cell of its direct neighbours during network construction. This effect is shown in Figure 5.13, where the Hong Kong traces are re-run with the network first *trained* by initially running the user through the system, to allow each cell to learn of its neighbours as contained in the trace data. This training provides a somewhat unfair advantage for a dynamic scheme, and hence these results are not claimed to be indicative of a precise real-world implementation. They do serve however to illustrate the lack of a low-call-rate performance advantage for static schemes once cells are provided enough information to learn of their general topology.

5.6 Summary

The performance of dynamic location management schemes has been shown to significantly improve over existing static systems for a wide range of network and user parameters. In the 'worst-case scenario' of pure homogeneity in network and user characteristics, or unrealistically low movement and call rates, the performance of dynamic schemes is seen to degrade simply to that of an ideal static scheme. For all realistic network formulations the performance of dynamic schemes is seen to be highly stable, capitalising on network variation where available.

Moreover the performance of a per-user dynamic scheme offers far greater cost reduction than a simple aggregate dynamic mechanism. For realistic networks containing a high degree of per-cell and per-user heterogeneity, a fully dynamic per-user scheme is able to characterise these aspects with a minimum of network information, providing significant cost reductions. Again this performance degrades to that of an aggregate dynamic scheme where no heterogeneity is present.

As network density increases in future networks, the subsequently increased movement rates provide a far finer granularity of information to the network for characterising user and cell parameters. With this increased movement rate the performance differential between static, aggregate dynamic and per-user dynamic schemes diverges by an even greater degree, as indicated by all simulations presented in this chapter. Such a trend in cost reduction offers great promise for increased benefit from a per-user dynamic LM scheme in future *picocell* networks, accentuating the importance of per-user dynamic location management as time progresses.

Chapter 6

Methods for LA Communication

Once the ideal location area for a user has been computed this information must be transmitted to the user device. Static location update schemes require no such communication as the LA is defined implicitly by the network and observed directly by the device. Dynamic schemes however, such as that proposed in Chapter 3, require a network controller to calculate this LA and send it to the client device as part of the location update process. Care must hence be taken to ensure the additional communication overhead involved is minimised, to avoid negating the efficiency gains of an idealised location area.

While low transmission size is the primary goal in the decision on an LA communication method, accuracy also requires careful consideration. A compromised location area may be chosen in the aims of reducing the size of the LA data representation, with an inherent trade-off in LA optimality. It is also vitally important for the network and device to have a consistent record of the location area, to avoid a device becoming ‘lost’ when not able to be found via paging in the network’s record of the device’s locality.

Implementation in legacy systems raises additional limitations in storage on the mobile device. For widespread implementation in current GSM systems, all location information must be stored on the user’s SIM card, affirming the importance of a terse representation of the cells within the current location area.

A number of methods for LA communication are proposed in the following sections, with the Bloom Filter representation of Section 6.4 proffered as the ideal solution for the majority of networks.

6.1 Direct List

A direct list of the cells contained within a location area is the simplest and most intuitive data representation. In addition to its computational simplicity, both for network and device, a direct list of cells offers zero error rate, an appealing characteristic for both analysis and implementation.

Unfortunately the uncompressed representation entails a large communication overhead, particularly for very large location areas, as will likely be common in future networks. While the cost of transmission of a large LA is somewhat mitigated by the subsequently infrequent location update rate, the poor scalability represents a significant cost penalty where bandwidth minimisation is a primary concern. Infrequent updates also offer no concession against device storage requirements, and hence location area size is limited significantly by available storage on the mobile device.

In addition to poor scalability, the transmission size of each list is directly related to the number of cells in the LA, and thus highly variable. This characteristic poses implementation challenges, preventing asynchronous transmission and requiring additional communication overhead.

6.1.1 Scalability

The spatial scalability of a direct list scheme is clearly O(n). It is important however to evaluate the communication cost of the scheme in absolute terms. The GSM cellular networks examined in Section 4.1 exhibit cell IDs ranging to 65513 (contained in location area 1024 of the Orange Paris network), indicating a current 2 byte limit on cell IDs. Each cell broadcasts an additional 2 byte identifier, corresponding to its location area code. A dynamic LM scheme would render such a location area identifier redundant, and with an expected growth in the number of cells, it is proposed these two identifiers are merged into a single 4 byte cell ID. With such a scheme the communication cost, ignoring synchronisation overhead, equates to $4n$ for a n-cell location area.

6.1.2 Sample Representation

Figure 6.1 shows a dynamically generated location area for a user performing a location update in cell 24952 of the Telstra Sydney GSM network. The location area is overlaid on a map of north-west Sydney [Pub04], illustrating the general location of cells to provide a geographical reference frame for the representation scheme of Section 6.2. The greyed cells outside of the location area are immediately adjacent to those within the indicated area, as recorded by the network based on previous user movements. Upon movement to any of these cells the user should register a departure from the current location area. This sample LA will be used in providing representations for all communication methods presented here.

The direct list representation for the sample location area is given in decimal form as:

```
1311 1312 1323 6643 7421 7422 7423 7771 8051 9501 9702 9752 10672
10771 10781 10783 14162 24951 24952 24955 25161 25162 25165
```

for a total of 46 bytes in a 2-byte representation or 92 bytes in a 4-byte per cell ID scheme.

With a low call arrival rate, a location area may be much larger. For a user receiving only one mobile phone call every two days, location areas up to 70 cells large were recorded in the Sydney Telstra network traces. This equates to a communication overhead of 280 bytes, significant when compared to a current communication cost of only a few bytes.

6.2 Polygon Approximation

Traditional attitudes towards location management have closely associated user movement with geographical coordinates. This has led to a concept of location area in terms of physical

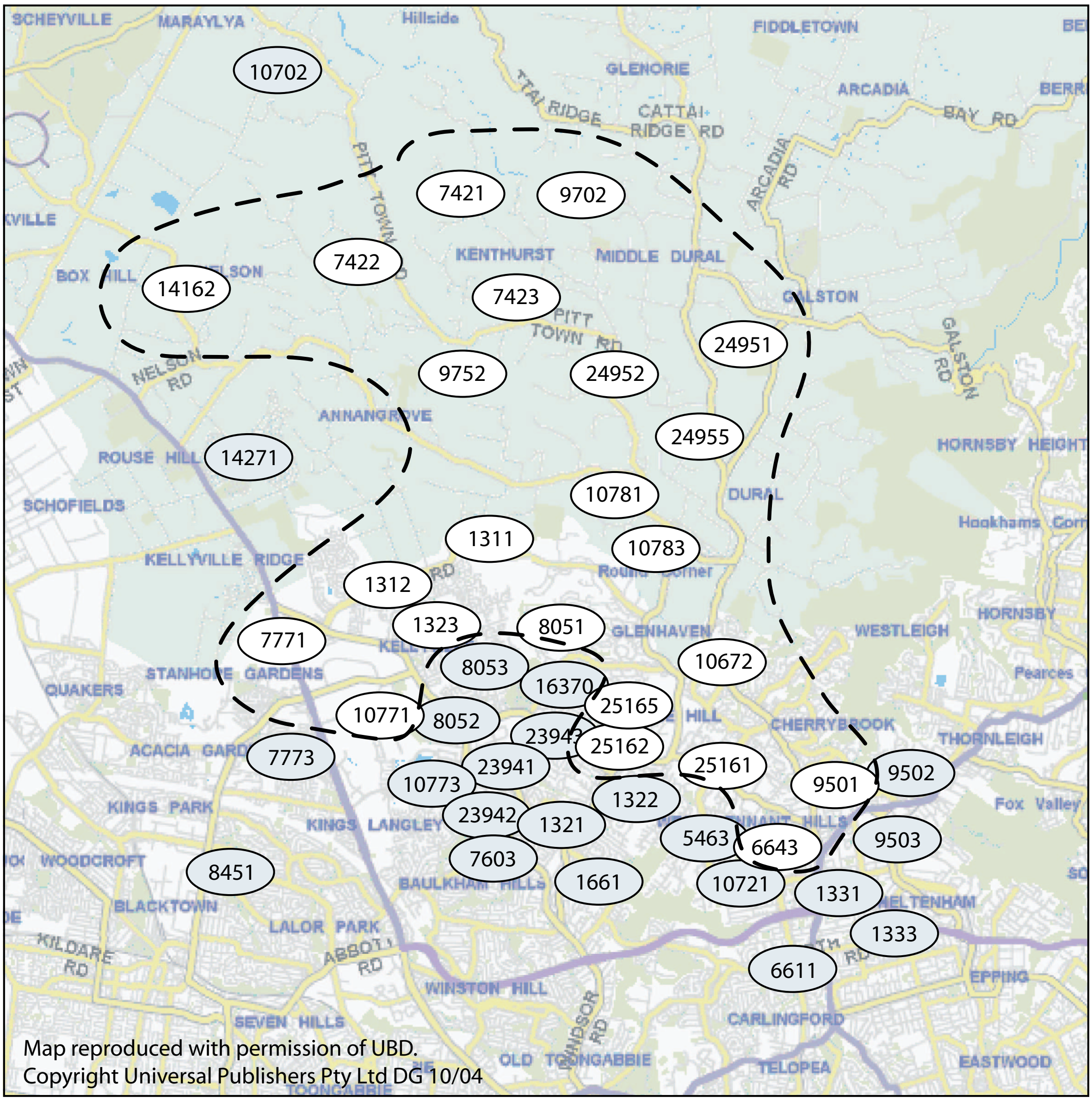


Figure 6.1: Sample dynamically-generated location area for user at cell 24952 in Telstra Sydney GSM network [Pub04]

space. When network topology is removed from the unrealistic assumptions of a perfect hexagonally-tessellated cell pattern, with location areas no longer approximating a circle centred around the current user location, such description in terms of a coordinate system become more difficult.

One method of representing a location area in terms of physical space would be to approximate it by a polygon, requiring the network to send only the coordinates of each vertex to the user. This method is well suited to traditional dynamic location management schemes where each location area is of either circular or elliptical aggregate shape. With a more comprehensive location management approach however, as is presented here, a location area need not conform to such simplistic descriptions. When tailored to network movement characteristics each location area may be of arbitrary complexity, containing concave regions and the possibility of ‘holes’ or hollow regions in the location area - for example, a cell within a building that a user is highly unlikely to enter. Such complexities make the designation of an optimum polygonal approximation exceedingly difficult with a limited number of vertices, or impossible in the instance on holes within the LA. Section 6.2.2 demonstrates this complexity for even a relatively simple location area.

While a simple polygon-inclusion test would allow devices to efficiently determine the occurrence of a departure from their location area, the main disadvantage is the computational complexity involved for the network in fitting such a polygon around an arbitrary set of cells. Any processing of the location area representation must be kept to a minimum to allow any feasible implementation in a network with millions of concurrent users. With no known polynomial time algorithm for such a computation, as well as no guaranteed error rate, the definition of a location area in terms of physical coordinates can be seen as an unwise decision and a vestige from previous location management philosophies.

6.2.1 Scalability

The main advantage of a polygonal approximation is the resultant small transmission size. If the number of vertices in each polygon is fixed, as would likely be required for efficient implementation in devices, this transmission size is also constant, allowing a further reduction in synchronisation overhead.

3 bytes for each geographical coordinate would allow close to 200mm accuracy over a network as large as the entirety of Australia. An additional byte would provide for better than 10mm possible resolution over the entire globe with a simple latitude/longitude grid mapping. Adopting the former, each location area can be represented trivially at a constant O(1) cost of $2 \cdot 3 \cdot n = 6n$ for an n dimensional polygon, or simply 36 bytes for a hexagonal system.

6.2.2 Sample Representation

Figure 6.2 shows a hexagonal approximation to the location area given in Section 6.1.2. This approximation includes all cells in the LA, their physical coordinates defined by their cell centre, but erroneously includes neighbouring cells 14271, 16370 and 1322. While the inclusion of 3 redundant cells in a 23 cell location area is a small penalty, the error factor would be much greater given a dense cell configuration in the vicinity of cell 14271, or a typically larger LA in a widely dense network.

6.3 Data Compression

A casual glance at the location area of Section 6.1.2 reveals a large degree of data redundancy. Current cell naming conventions prefix cells originating at the same physical transmitter with the same sequence of digits, such as cells 22931, 22932 and 22933, all located in Carlingford in the Telstra Sydney GSM network. This redundancy may be easily exploited in reducing the size of the LA information transmitted to the device.

Simple compression techniques such as Huffman coding [Huf52] or LZW compression [Wel84] may be used to exploit such redundancy. As only the last digit in a cell ID varies across cells in a given physical area, this last digit may be treated as a separate symbol from the first two - four digits, allowing Huffman coding to be performed over the resultant symbols. LZW compression may treat each digit in a stream of cell IDs as a separate symbol, building a

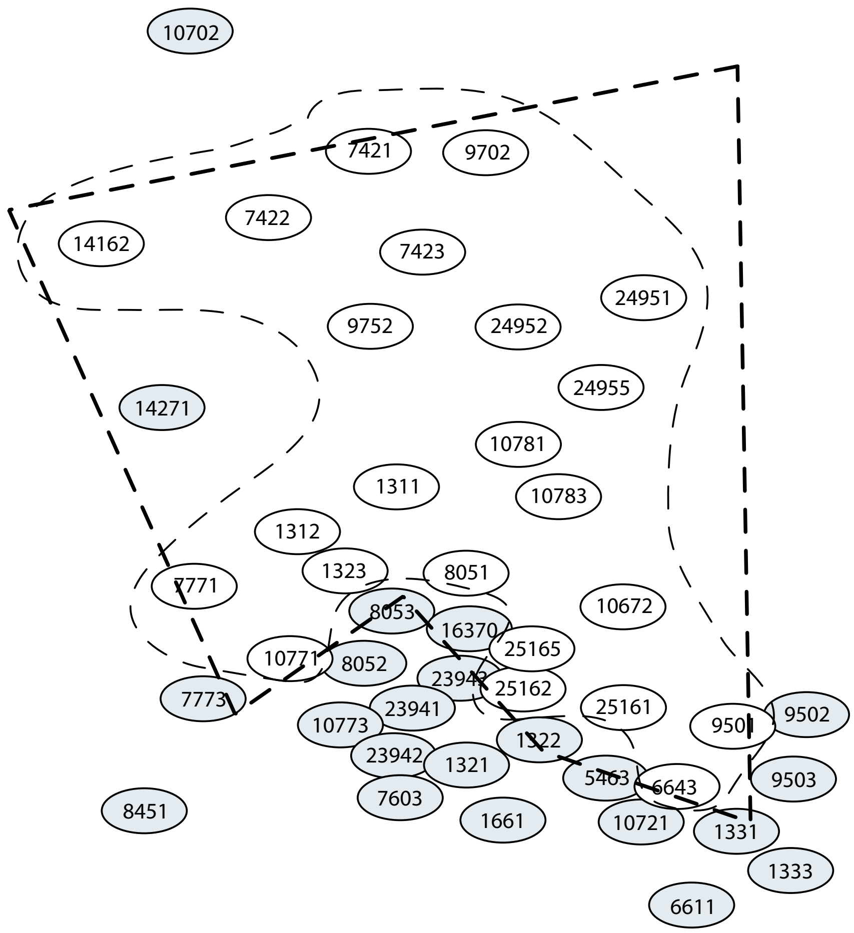


Figure 6.2: Sample location area overlaid with hexagonal polygon approximation

symbol table as they are scanned. While performance gains are possible in both schemes, the location area must be sufficiently large to mitigate the cost of transmitting the symbol table along with the compressed LA data.

A compression scheme based on cell IDs also relies on a degree of regularity in names within a region. While a common phenomena in current networks, this places heavy restriction on naming schemes used by operators, and limits the opportunity for adding cells to a region where utilisation dictates. This reduction in flexibility is a significant hindrance to network expansion and counter-intuitive to the notion of network abstraction. The burden placed on operators in such a scheme, as well as a bias against future network expansion, controverts the gains achieved via a reduction in communication overhead.

Regardless of compression performance, the transmitted data must be decompressed by the mobile device and stored in an uncompressed form, to allow low computational overhead. Hence a compression scheme in isolation offers no reduction in device storage requirements while requiring the implementation of efficient decompression on the mobile device.

6.3.1 Scalability

As most cell crossings are the result of hopping between cells covering the same physical region, it is highly likely that the three cells located at a single transmitter will be included in a given location area. This will result in multiple occurrences of 2 or 3 cell IDs differing only in their last digit, as demonstrated in Section 6.1.2. Additionally, these last digits have a limited range, in almost all cases either 1, 2 or 3, across all the GSM networks surveyed. For a large location area we may expect a very high occurrence of single digits 1, 2 and 3, with each 3-4 digit prefix occurring between 1 and 3 times. Such constrained network descriptions signal a relatively high level of redundancy, as demonstrated in the following section.

The precise compression performance is dependent on the specific compression scheme utilised and the redundancy inherent in each location area. Unfortunately as location area size increases the probability of a given 3-4 digit prefix occurring decreases compared with the probability of each single digit, leading to an increase in entropy according to basic coding

Symbol	"121"	"1"	"2"	"3"	"742"	"1078"	"2495"	"2516"	"5"	others
Multiplicity	2	9	8	4	3	2	3	3	2	1

Table 6.1: Symbol likelihoods for sample location area

theory. This increased entropy points to a less compressible dataset, and subsequent poorer performance. Since compressibility degrades over an increase in LA size, the scalability of such a representation can only be claimed as a linear O(n).

6.3.2 Sample Representation

If the LA of Section 6.1.2 is segmented into cell ID prefixes and trailing digits we have a total of 46 symbols, with multiplicities given in Table 6.1. The entropy for this system is given by $H = \sum -p(i) \cdot \log_2(p(i))$ where $p(i)$ is the probability of each symbol i occurring in the location area. This entropy evaluates to 3.77 in this case, indicating a best-case representation requiring an average 3.77 bits per symbol using a Huffman-like compression scheme. Again assuming ideality we may expect 174 bits in total to represent the compressed location area. While this 22 bytes of data is slightly less than half that required in the direct list representation, it must be understood that this figure is an ideal case only, nor does it include the communication overhead required in transmitting the symbol table. With the additional communication and computational overhead, a simple compressed representation becomes a far less appealing option for LA communication.

6.4 Bloom Filters - Towards true abstraction

An ideal LA communication method would allow a highly compressed representation of location area information while minimising the amount of computational overhead required by the device in interpreting this data. It is also highly desirable to keep this information in compressed form on the device, to avoid limitations resulting from device storage capacity. As it is important for the network to minimise the aggregate LM cost, as opposed to the

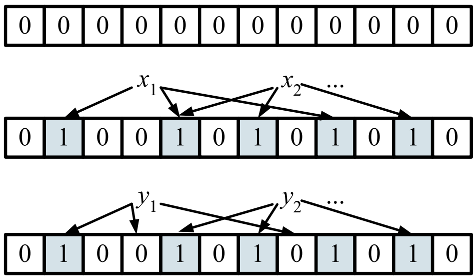


Figure 6.3: Bloom filter example [BM02]

cost for each individual device, it may also be acceptable to tolerate a small degree of error, manifested in a less ideal location area.

The Bloom filter data structure [Blo70] offers a compressed data representation tailored precisely to the needs of efficient set membership queries, while requiring a minimum of storage space. This efficiency is offered at the penalty of a small probability of false positives - where an element not contained within the set is incorrectly identified as being a set member. While previously consigned solely to the domain of database optimisation, Bloom filters are now garnering significant attention throughout the network community [BM02].

6.4.1 Bloom Filter Fundamentals

A Bloom filter is an array structure of length m bits used to represent a set $S = \{x_1, x_2, ..., x_n\}$ of n elements. An additional variable k defines a number of hash functions $h_1, h_2, ..., h_k$ uniformly mapping the universe of values to an index into the m-array. Each element $x_i \in S$ is hashed according to the k hash functions, with each corresponding index in the array set to 1. This process is demonstrated for $m = 12$, $n = 2$ and $k = 3$, with the following hash values, in Figure 6.3.

$$h_1(x_1) = 4, \quad h_2(x_1) = 1, \quad h_3(x_1) = 8$$
$$h_1(x_2) = 10, \quad h_2(x_2) = 6, \quad h_3(x_2) = 4$$

As shown in Figure 6.3, each bit in the array is initialised to zero. Each index is then progressively set to 1 for each hash value, remaining unchanged if subsequent hashes correspond to the same index. To check for existence of an element, the element is hashed by each of the k functions and the corresponding indices examined in the array. If any of these array positions have not been set to 1, as is the case with y_1, then the element does not exist in the original set. If not, we may say with high probability that the element was contained within the original set, as demonstrated by y_2. There is a distinct possibility however that a false positive will arise, such as the test for existence of an element hashing to indices 1, 4 and 10 in this case - while all of these are set to 1 in the diagram, no such element existed in the input set.

The false positive rate is central to the effectiveness of a Bloom filter implementation and, given a sufficiently random set of hash functions, may be approximated as [BM02]:

$$p = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k \tag{6.1}$$

This error rate is subsequently minimised with the number of hash functions chosen such that $k = \ln 2 \cdot (m/n)$. When substituted into Equation 6.1 we arrive at an idealised approximate false positive rate of:

$$p = (0.6185)^{m/n} \tag{6.2}$$

As n is essentially a fixed quantity in a given implementation, it is clear that an appropriate selection of m is vitally important in the design of an effective Bloom filter, and determines the trade-off between spacial efficiency and accuracy. In an ideal system an m/n ratio, the average number of bits per element, of 4 will give a false positive rate of 14.6% while an 8 bit-per-element filter will give a much less significant 2.1% error probability.

6.4.2 Application to LA Communication

Bloom filters seem initially to apply very well to the task of LA communication. They meet exactly the requirements of small data representation coupled with the provision of highly efficient existence tests on this compressed form. This offers significant gains over a compression-based scheme as storage requirements are minimised, not just transmission overhead. For large location areas the computational complexity is lower than even a direct list scheme, with constant-time lookup to check for existence of a cell within the LA, as opposed to linear in the list representation, with respect to the number of cells in the location area.

The hashing of cell IDs to k array indices also abstracts the length of the cell name away from device storage requirements, allowing a greater flexibility in naming schemes. This will become ever more important as network density and heterogeneity increases, signalling a need for an unrestricted naming system.

One weakness requiring consideration however is the possibility of false-positives when determining if a given cell is represented within a given Bloom filter. A naïve implementation of Bloom filters in communicating LA data may have rather severe consequences. If a device moves out of its current location area but mistakenly believes that its new cell is still within the LA prescribed by the network, as would be the case for a false-positive, the device will not signal a location update, essentially becoming 'lost' within the network. Here the network incorrectly assumes that the device is contained within the paging area defined by the LA, and will be unable to find the device on an incoming call. This collapse of network stability is clearly unacceptable in a modern cellular network, yet is thankfully easily remedied with careful implementation.

From Figure 5.1 in the previous chapter, it can be observed that the cost penalty for a larger-than-necessary location area is comparatively small, exhibited by the low gradient at the tail of the cost plot. We may hence make the hypothesis that an increase in location area size has a minimal impact on the true cost imparted on the network. This environment allows us to make a small increase in the size of each location area to ensure a perfectly consistent view

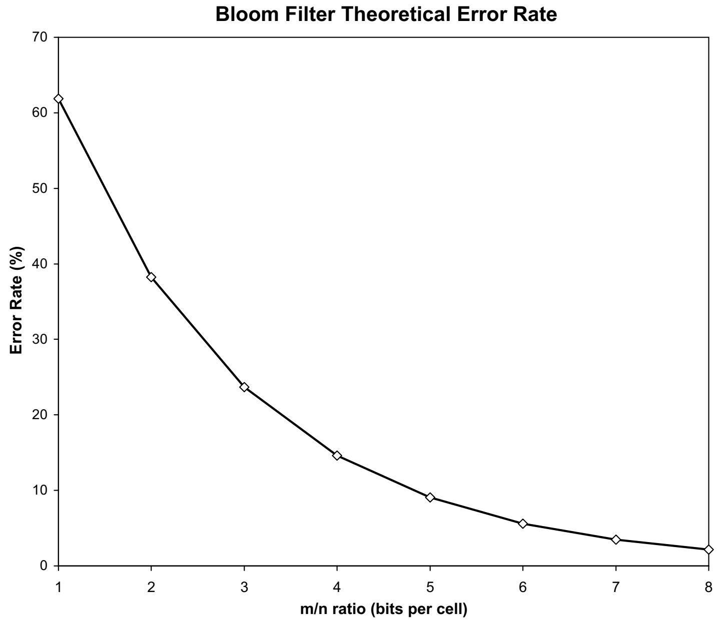


Figure 6.4: Theoretical error rate for an ideal Bloom filter implementation

from both the network and device perspective, in effect eliminating the possibility of a false positive. Here once the network has produced the Bloom filter representation for the optimum location area for a user, it will then add any neighbouring cells to the area if they result in a false-positive. If this process is repeated until no adjacent false-positive cells remain, there will be no possibility of a device unknowingly leaving its location area. Additionally, as the penalty for a larger location area is very low, much higher theoretical false-positive rates may be tolerated while ensuring adequate performance.

6.4.3 Performance

Figure 6.4 shows the theoretical false-positive rate for an ideal Bloom filter, across a feasible range of m/n ratios. Here it is assumed that 8 or less bits per cell ID will give a significant saving in communication overhead. It can be seen that for less than 4 bits the false-positive rate rises above 20%, an apparently unacceptable figure when analysed in isolation. When combined with the knowledge that the cost penalties for a larger location area are quite small however, this error becomes a far more feasible proposition.

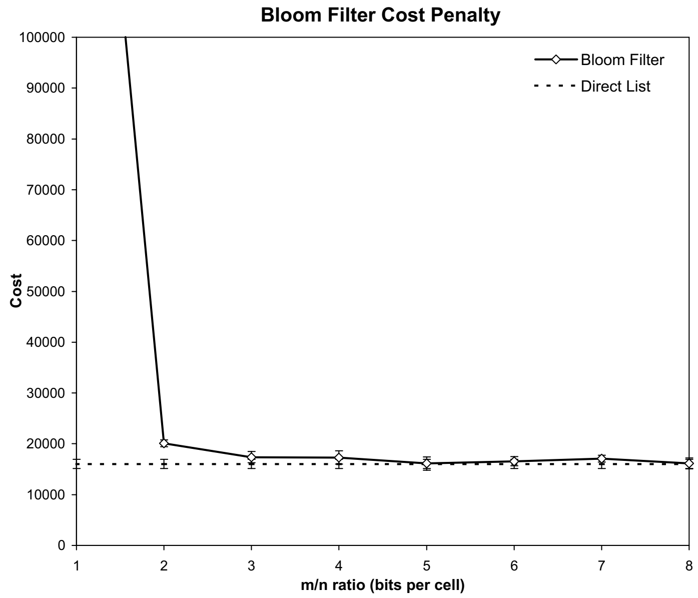


Figure 6.5: LM cost for Bloom Filter and Direct List representations on Telstra Sydney GSM network

Figure 6.5 presents the location management cost incurred when using Bloom filters to communicate location area data as generated in the Telstra Sydney GSM network traces of Section 4.1. Here the size of the Bloom filter in bits is set as a given multiple of the number of cells required in each LA communication, rounded to the nearest byte. It should be noted that the average LA size in these traces is 49 cells, the large value reducing any artifactual effects of the rounding. It can be observed that the cost penalty for a Bloom filter implementation is very low for even small values of m/n - for a Bloom filter of length 4 bits per cell this penalty is near negligible.

Analysis of the communication overhead presented in Figure 6.6 reveals the true benefits of the use of Bloom filters. Here for the same 4 bits per cell, the LA communication overhead is a mere 25% of that otherwise required for a direct list implementation. It is important to note that these savings not only impact on the bandwidth required for location management, but also alleviate the storage and processing requirements of mobile devices.

Armed with proprietary knowledge of specific location management packet formats and sizes, a trade-off may be readily made between LM cost and communication overhead, minimising

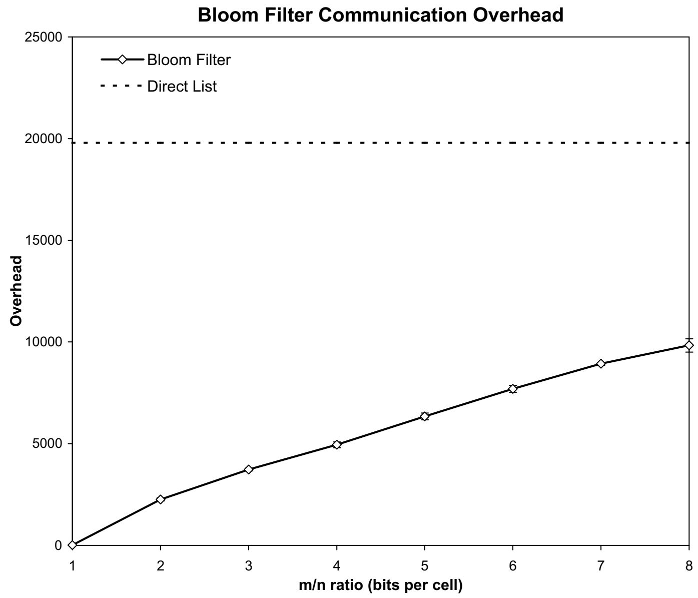


Figure 6.6: LA communication overhead for Bloom Filter and Direct List representations on Telstra Sydney GSM network

the total cost for telecommunications operators. With this information unavailable however it may still be clearly observed that even for a relatively large m/n ratio of 4, abstract LM cost is increased by a minimal to negligible level while the specific cost of LA communication is reduced by a factor of 4. As net bandwidth cost incurred by operators relies on not only the total number of location updates, but also the size of each individual update message, it is hence possible for operators to fully optimise their bandwidth use through the use of Bloom filters and an appropriate selection of m/n ratio.

6.4.4 Scalability

The spatial scalability of the Bloom filter implementation depends on whether the size of the filter is fixed or set to be a factor of the number of cells in the location area. If there is low variability in LA size then the former option may be adopted, with a fixed Bloom filter length m and varying level of cost penalty. Here the scalability of the scheme is simply a constant O(1). It is suggested for more heterogenous networks that m is selected according to the number of cells in a given location area, maintaining a fixed cost penalty but varying

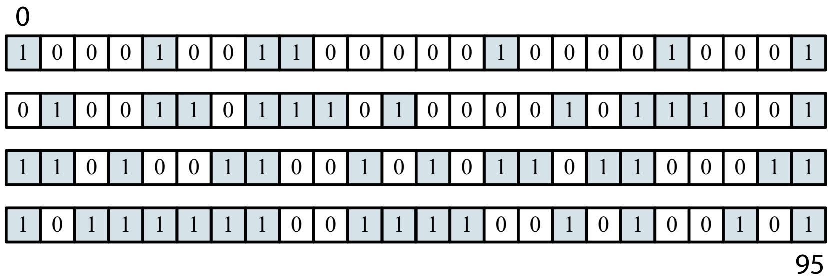


Figure 6.7: Sample Bloom filter representation

communication size. Here transmission length is O(n), with a low coefficient corresponding to the small number of bits required per cell.

6.4.5 Sample Representation

The 23 cells in the location area of Figure 6.2 would require a Bloom filter of length $m = 23 \times 4 = 92 \approx 96$ bits, rounded up to the nearest byte, with an m/n ratio of 4. An optimal number of hash functions in this case is $k = \ln 2 \cdot (m/n) = 4 \cdot \ln 2 = 2.77 \approx 3$. Sample hashes for each of these cells, along with their immediate neighbours, are given in Appendix C. Figure 6.7 illustrates the 12 byte Bloom filter representation of this LA.

Cell 9502 is the only cell that results in a false positive in the Bloom filter - its hashes 44, 48 and 93 all set to 1. The network would hence add this cell to the location area and examine 9502's neighbours for false positives, continuing this process. For simplicity here we assume that none of the neighbours of cell 9502 result in a false reading. If the device moves out of the location area into cell 1661 for example, it will hash this cell's identifier, examine the corresponding indices 27, 56 & 66 in the Bloom filter and, finding cells 27 and 56 set to zero, deduce that cell 1661 is not a member of the previous location area.

6.5 Summary

A number of methods for communicating a dynamic location area to a user are proposed and analysed here. While not addressed in previous research, it is essential that such an investigation be conducted, to ensure that the additional communication required in transmitting a complex location area does not mitigate the performance gains from a dynamic system. A Bloom filter representation was found to be ideal for the purposes of LA communication, allowing a highly compressed representation, efficient computation and minimal restriction on location area formation.

Chapter 7

Future Work

The location management scheme proposed here is a significant departure from current research and signals a wide range of future work. Much of this is scheduled for completion in the near future with continual developments to progress for a significant period of time. It is expected that this research will form the basis of a series of publications, ideally signalling a new direction for research into location management techniques.

7.1 Timer-based Location Update

The possibility of enforcing an additional timer-based location update scheme, in parallel with the proposed system, deserves consideration. The current scheme assigns a new location area to a user only when they leave their existing LA. While intuitively appropriate, such a scheme does not accommodate for a very small subsection of users with unexpected movement characteristics. An example of this would be a user who is moving very frequently, perhaps inter-state, and hence is assigned a very large location area, but then ceases this frequent movement and remains in their local area for an extended period of time. In such a situation the user may never leave their location area and will remain consigned to their inappropriately large LA, placing heavy paging load on the network for every incoming call. A mandatory update after a given period of time would allow the network to reassign a smaller location area to such users, yet the merits of this need to be considered against the unnecessary updates that will be triggered for more typical users.

7.2 Reduced Significance of Movement Factor

As network density increases, the relation between physical geographical movement and logical movement between individual cells becomes increasingly decoupled. While the movement *factor* characterisation of Section 3.3.2 avoids the fate of previous schemes in the presence of ping-ponging and cell-hopping movements, an investigation into the continued relevance of individual user movement in location management would yield interesting results. Although user movement will always be a significant factor in a 'perfect' location management scheme, the gains available from assuming all users move with the same speed are significant - the matrix computations of Section 3.5 need be executed with greatly reduced regularity, owing to a constant user movement rate and less frequently varying parameters. An evaluation of the reduction in computational cost, contrasted with the associated reduction in prediction accuracy, may raise possibility of implementing a 'compromised' scheme to reduce outlay in required infrastructure.

7.3 Modern Cellular Network Trace Package

A major hindrance for the evaluation of a realistic dynamic location management scheme is a lack of available trace data. Without trace data capturing the complex movement behaviour of individual users, with realistic call and movement rates, it is very difficult to evaluate the true performance of a per-user dynamic LM proposal. It would be highly desirable to extend the trace data recorded in Chapter 4 to a full set of realistic cellular network traces. Such traces would not only ascertain the performance of the given LM scheme with greater confidence, but would also remove the ambiguity involved in comparing LM schemes tested with highly subjective simulation environments.

7.4 Evaluation of $\varsigma_{LU} : \varsigma_P$ Ratio

As detailed in Section 3.7, while the ratio between location update cost and paging cost is estimated as 10:1, there is little theoretical evidence to support this assumption. It is suspected in fact that the real ratio between these cost factors is much higher, as indicated by the very large static location areas currently implemented in modern GSM networks. With the added computational cost during location update incurred by a dynamic LM scheme, previous estimates of $\varsigma_{LU} : \varsigma_P$ ratio are likely to be significantly inaccurate. It would be of great theoretical interest to determine a realistic ratio between LU and paging cost for modern networks, to more appropriately compare simulated performance of LM proposals with currently implemented static schemes. The isolation of this data would require assistance by a major telecommunication provider, as well as access to previously unpublished information, and is hence a task for collaborative effort rather than an individual research goal.

7.5 Optimised Prototype Implementation

A great number of possible optimisations are proposed throughout Chapter 3, offering significant potential increases in performance. Many of these optimisations were not implemented in the final simulation framework to allow a clearer evaluation of the effect of various simulation parameters, and to aid in debugging and testing of the LM scheme while in development. A more comprehensive evaluation of feasibility for real network implementation requires the development of a full location management prototype, free from simulation overhead and capable of receiving external input from ‘users’ to define network activity. Such a prototype would allow the precise evaluation of system performance in terms of throughput and latency, proving a concrete measure of the computing power required in implementing the LM scheme in a modern network.

7.6 Low-level Communication Description

With the completion of other further work, signalling a complete and feasible location management scheme, a low level description of the system is required. Such a specification would define communication in terms of precise packet headers, message formats and fully quantified location management overheads. Additional requirements for implementation, such as the computing power required by a central network controller, as well modifications necessary to SIM cards for each device in the network, need to be precisely established and justified in terms of financial outlay for network operators. Such work would again require collaboration with network providers, and hence feasible only in the event of serious interest from industry bodies.

7.7 Applicability to Future Networks

While the focus of the research presented in this document has been on the performance of dynamic location management in cellular mobile phone networks, particular those under the GSM architecture, the longevity of such systems may be seen as somewhat limited. Cellular communication systems of the not-too-distant future will likely see the convergence of mobile phone and personal computer wireless systems into a single common wireless infrastructure. The applicability of current location management schemes onto Mobile IP systems has already been demonstrated [ZCC02]. The dense *picocell* topologies of such convergent cellular communication systems signal great potential for the use of dynamic location management in optimising cellular networks of the future. A precise evaluation of the applicability of the proposed location management scheme to emerging cellular technologies would warrant continued development and implementation over an extended period of time.

Chapter 8

Conclusions

The proposed dynamic location management scheme has been shown to satisfy the goals prescribed at the commencement of this project, formalised in Section 1.3. This per-user dynamic scheme reduces location management cost over both existing static and idealised aggregate dynamic schemes, across a wide variety of user and network parameters. In addition this scheme has been shown to degrade to an optimum static or aggregate dynamic scheme where significant heterogeneity is not present, emphasising the stability of the per-user dynamic mechanism.

The weighted graph network abstraction allows application of location management to arbitrary cellular networks, both in theoretical homogeneous and realistic heterogeneous topologies. The lightweight probability and rate learning mechanism allows the abstraction to optimally represent a network at any point in time, capturing temporal characteristics and alleviating network operators from time-consuming and inaccurate manual network parameterisation.

The cost metric used in minimising overhead considers the simplistic parameters of user call arrival rate, user movement factor, cell dwell time and aggregate movement probabilities. These four quantities allow the assignation of a location area based on both user characteristics and cell structure within a location area - tailoring location management to user and network with a minimum of computational overhead. The overhead of the two computationally expensive operations of cell order determination and LA residence time estimation are analysed and found to be within the the capabilities of consumer computing hardware, avoiding the necessity of extreme computing power in the implementation of such a system.

Importantly, no assumptions are made with regards to regularity of user behaviour or uniformity of network topology. The proposed dynamic scheme is able to achieve high levels of performance for realistic as well as completely random user movement, on similarly varied network topologies.

The examination of specific performance involved the survey of current GSM networks, recording trace data and evaluating the relevance of existing cellular network benchmarks. The subsequent performance evaluation against real network traces raises the significance of the work from an extension of previous dynamic schemes, to a unique branch into feasible, provable, location management performance.

Thorough investigation is also given into the communication of dynamic location area information, with a Bloom filter approach suggested for maximum efficiency. This investigation rounds the development of a complete scheme for dynamic location management, maintaining regard to net cost reduction through the implementation of such a system.

As network density continues to increase, with cell-transitions becoming further decoupled from physical user movement, the performance of an abstract per-user dynamic scheme will continue to diverge away from traditional physical-based and aggregate location management proposals. The significance of this research however lies not only in providing a scheme offering greater location management performance, but in establishing a benchmark for practical, provable and implementable results in realistic networks, a far cry from hypothetical schemes prevalent in the current research community.

Bibliography

[ADM98] L.P. Araujo and J.R.B. De Marca. A comparative analysis of paging and location update strategies for pcs networks. In *Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on*, volume 3, pages 1395–1399 vol3, 1998.

[AHL96] I.F. Akyildiz, J.S.M. Ho, and Yi-Bing Lin. Movement-based location update and selective paging for pcs networks. *Networking, IEEE/ACM Transactions on*, 4(4):629–638, 1996.

[ALLC00] I.F. Akyildiz, Yi-Bing Lin, Wei-Ru Lai, and Rong-Jaye Chen. A new random walk model for pcs networks. *Selected Areas in Communications, IEEE Journal on*, 18(7):1254–1260, 2000.

[AMH+99] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioglu, and Wenye Wang. Mobility management in next-generation wireless systems. *Proceedings of the IEEE*, 87(8):1347–1384, 1999.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[BM02] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. In *Allerton Conference on Communication, Control, and Computing. 40th Annual Proceedings.*, pages 636–646, 2002.

[BMJ+98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97. ACM Press, 1998.

[BNIM95] A. Bar-Noy, I.Kessler, and M.Sidi. Mobile users: To update or not to update? *Wireless Networks*, 1(2):175–185, 1995.

[BNK93] A. Bar-Noy and I. Kessler. Tracking mobile users in wireless communications networks. *Information Theory, IEEE Transactions on*, 39(6):1877–1886, 1993.

[CBCS02] G. Chakraborty, B.B. Bista, D. Chakrabort, and N. Shiratori. Location management in pcn by movement prediction of the mobile host. In *Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on*, volume 1, pages 78–83 vol1, 2002.

[CLSS00] J. Chan, B. Landfeldt, A. Seneviratne, and P. Sookavatana. Integrating mobility prediction and resource pre-allocation into a home-proxy based wireless internet framework. In *Networks, 2000. (ICON 2000). Proceedings. IEEE International Conference on*, pages 18–23, 2000.

[CNLB92] R.S. Wolff C. N. Lo and R. C Bernhardt. An estimate of network database transaction volume to support personal communications services. In *Universal Personal Communications, 1992. ICUPC '92 Proceedings., 1st International Conference on*, pages 09.03/1–6, September 1992.

[Com04] Sony Ericsson Mobile Communications. P900 - specifications. `http://www.sonyericsson.com`, October 2004.

[CS04] Kuo-Hsing Chiang and N. Shenoy. A 2-d random-walk mobility model for location-management studies in wireless networks. *Vehicular Technology, IEEE Transactions on*, 53(2):413–424, 2004.

[CT02] Woo-Jin Choi and S. Tekinay. Distance-based location update scheme with isotropic random user motion with drift. In *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, volume 3, pages 1111–1115 vol3, 2002.

[EGO02] P.G. Escalle, V.C. Giner, and J.M. Oltra. Reducing location update and paging costs in a pcs network. *Wireless Communications, IEEE Transactions on*, 1(1):200–209, 2002.

[EIA91] EIA/TIA. Cellular radio telecommunications intersystem operations. Technical report, July 1991.

[FI03] B. Furht and M. Ilyas. Wireless internet handbook - technologies, standards and applications. 2003.

[Fis04] A. Fischer. Celltrack. `http://www.afischer-online.de/sos/celltrack/`, October 2004.

[Gon96] P. R. L. Gondim. Genetic algorithms and the location area partitioning problem in cellular networks. In *Vehicular Technology Conference, Mobile Technology for the Human Race, Proc. IEEE 46th*, volume 3, pages 1835–1838, April 1996.

[GR98] Daqing Gu and S.S. Rappaport. A dynamic location tracking strategy for mobile communication systems. In *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE*, volume 1, pages 259–263 vol1, 1998.

[Gra04] Zero Gravity. Symbian uiq missing files. `http://zerogravity.narod.ru/symbian/uiq/etelmm.html`, October 2004.

[Gro04] Nokia Group. Nokia - nokia n-gage. `http://www.nokia.com`, October 2004.

[Gup02] Anshul Gupta. Recent advances in direct methods for solving unsymmetric sparse systems of linear equations. *ACM Trans. Math. Softw.*, 28(3):301–324, 2002.

[HCT00] Hang-Wen Hwang, Ming-Feng Chang, and Chien-Chao Tseng. A direction-based location update scheme with a line-paging strategy for pcs networks. *Communications Letters, IEEE*, 4(5):149–151, 2000.

[Huf52] David A. Huffman. A method for the construction of minimum-redundancy codes. In *Proc. IRE*, volume 40, pages 1098–1101, September 1952.

[Kle01] Paul Klemperer. How (not) to run auctions: the european 3g telecom auctions. `http://www.nuff.ox.ac.uk/users/klemperer/hownot.pdf`, November 2001.

[Kol03] Natalie Elizabeth Kolodziej. Location management in cellular networks using reliable metrics, 2003.

[KSS01] Thomas Kunz, Atif A. Siddiqi, and John Scourias. The peril of evaluating location management proposals through simulations. *Wirel. Netw.*, 7(6):635–643, 2001.

[Lee95] W.C.Y. Lee. *Mobile Cellular Telecommunications - Analog and Digital Systems.* McGraw-Hill, Inc., 1995.

[LH03] B. Liang and Z.J. Haas. Predictive distance-based mobility management for multidimensional pcs networks. *Networking, IEEE/ACM Transactions on*, 11(5):718–732, 2003.

[Lin97] Yi-Bing Lin. Reducing location update cost in a pcs network. *Networking, IEEE/ACM Transactions on*, 5(1):25–33, 1997.

[LJCW96] D. Lam, J. Jannink, D.C. Cox, and J. Widom. Modeling location management in personal communications services. In *Universal Personal Communications, 1996. Record., 1996 5th IEEE International Conference on*, volume 2, pages 596–601 vol2, 1996.

[LR98] Zhuyu Lei and C. Rose. Wireless subscriber mobility management using adaptive individual location areas for pcs systems. In *Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on*, volume 3, pages 1390–1394 vol3, 1998.

[Ltd04] Symbian Software Ltd. Symbian os - the mobile operating system. `http://www.symbian.com`, October 2004.

[NL98] Z. Naor and H. Levy. Minimizing the wireless cost of tracking mobile users: an adaptive threshold scheme. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 720–727 vol2, 1998.

[OOYM91] S. Okasaka, S. Onoe, S. Yasuda, and A. Maebara. A new location updating method for digital cellular systems. In *Vehicular Technology Conference, 1991. 'Gateway to the Future Technology in Motion', 41st IEEE*, pages 345–350, 1991.

[Phi95] S. Phillips. Telephone call traffic data set, 11/94-7/95, personal communication. Technical report, October 1995. Encrypted ID numbers.

[PLE04] PLEIADES. Sumatra: Stanford university mobile activity traces. `http://www-db.stanford.edu/sumatra`, October 2004.

[Pub04] Universal Publishers. Compound map of north-west sydney region, used with permission. UBD Sydney and Blue Mountains Street Directory, 41st Edition, 2004.

[Rah93] M. Rahnema. Overview of the gsm system and protocol architecture. *Communications Magazine, IEEE*, 31(4):92–100, 1993.

[Ros96] C. Rose. Minimizing the average cost of paging and registration: a timer-based method. *Wirel. Netw.*, 2(2):109–116, 1996.

[Ros99] C. Rose. State-based paging/registration: a greedy technique. *Vehicular Technology, IEEE Transactions on*, 48(1):166–173, 1999.

[RY95] Christopher Rose and Roy Yates. Minimizing the average cost of paging under delay constraints. *Wirel. Netw.*, 1(2):211–219, 1995.

[SK99] J. Scourias and T. Kunz. Activity-based mobility modeling: realistic evaluation of location management schemes for cellular networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 296–300 vol1, 1999.

[Soh03] Jan Sohnrey. Persönliche navigationshilfe auf basis von mobiltelefon-technologie. Master's thesis, Institut für Informatik der Universität Zürich, 2003.

[SZ01] R. Subrata and A.Y. Zomaya. Location management in mobile computing. In *Computer Systems and Applications, ACS/IEEE International Conference on. 2001*, pages 287–289, 2001.

[SZ03] R. Subrata and A.Y. Zomaya. Evolving cellular automata for location management in mobile computing networks. *Parallel and Distributed Systems, IEEE Transactions on*, 14(1):13–26, 2003.

[Tab97] S. Tabbane. Location management methods for third generation mobile systems. *Communications Magazine, IEEE*, 35(8):72–78, 83–4, 1997.

[TZss] J. Taheri and A. Y. Zomaya. Realistic simulations for studying mobility management problems. *International Journal of Wireless and Mobile Computing*, in press.

[Wel84] T. A. Welch. A technique for high performance data compression. *IEEE Transactions on Computers*, 17(6):8–19, 1984.

[WL00] V.W.-S. Wong and V.C.M. Leung. Location management for next-generation personal communications networks. *Network, IEEE*, 14(5):18–24, 2000.

[XTG93] H. Xie, S. Tabbane, and D.J. Goodman. Dynamic location area management and performance analysis. In *Vehicular Technology Conference, 1993 IEEE 43rd*, pages 536–539, 1993.

[YLN03] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 2, pages 1312–1321 vol2, 2003.

[ZCC02] Xiaowei Zhang, Javier Gomez Castellanos, and Andrew T. Campbell. P-mip: paging extensions for mobile ip. *Mob. Netw. Appl.*, 7(2):127–141, 2002.

Appendices

Appendix A

Cell ID Information

The real world trace data of Section 4.1 was recorded using consumer mobile phones, with software written for the Symbian operating system [Ltd04]. Initial investigations into cell sizes were performed using a Sony Ericsson P900 [Com04], while a Nokia N-Gage [Gro04] was used for the final gathering of trace data, owing to limitations in the Sony Ericsson UIQ API as discussed in Section A.2. The code used to obtain location information from a Symbian mobile phone is reproduced here to allow replication of the trace data results.

While cell information is available via C++ function calls to the Symbian OS, these calls require the use of unsupported function headers, removed from the release versions of vendor SDKs. Replacement headers, containing the requisite function definitions, are nonetheless in common circulation [Gra04] and required only for compilation off-device. The functions to obtain location information are dependent on the specific Symbian interface used by a device vendor, with the appropriate code listed below for Nokia and Sony Ericsson mobile phones.

A.1 Nokia Series 60

Both cell ID and location area code information is available on Nokia devices implementing the Series 60 user interface. An unrestricted version of the etelbgsm.h header file [Gra04] is required for compilation. The code to access cell ID and location area code data is given in

```
#include <etelbgsm.h>

void CClassName::GetLocationInfoL(TInt& cellID, TInt& LAC) {

    RTelServer server;
    User::LeaveIfError(server.Connect());
    User::LeaveIfError(server.LoadPhoneModule(_-
    L("Phonetsy.tsy")));

    RTelServer::TPhoneInfo info;
    User::LeaveIfError(server.GetPhoneInfo(0, info));

    RBasicGsmPhone phone;
    User::LeaveIfError(phone.Open(server, info.iName));

    MBasicGsmPhoneNetwork::TCurrentNetworkInfo networkInfo;
    User::LeaveIfError(phone.GetCurrentNetwork(networkInfo));

    cellID = networkInfo.iCellId;
    LAC = networkInfo.iLocationAreaCode;

    phone.Close();
    server.UnloadPhoneModule(_L("Phonetsy.tsy"));
    server.Close();

}
```

Table A.1: Code for obtaining current Cell ID and Location Area Code on mobile devices supporting the Nokia Series 60 SDK

Table A.1. This must be linked to both etel.lib and gsmbas.lib, available in the Nokia Series 60 SDK.

A.2 Sony Ericsson UIQ 2.x

While function calls exist to retrieve location area code from UIQ 2.x Sony Ericsson devices, this value is always returned as 0 on Sony Ericsson P800 and P900 devices. Cell ID data has been tested however and found valid on these phones. Here the complete etelmm.h header file [Gra04] is required, linked to both etel.lib and etelmm.lib. The code for obtaining the current cell ID, based on that presented in [Soh03], is given in Table A.2.

```
#include <etelmm.h>

void CClassName::GetLocationInfoL(TInt& cellID, TInt& LAC) {

    RTelServer server;
    User::LeaveIfError(server.Connect());
    User::LeaveIfError(server.LoadPhoneModule(_-
    L("ERIGSM.TSY")));

    RTelServer::TPhoneInfo info;
    User::LeaveIfError(server.GetPhoneInfo(0, info));

    RMobilePhone phone;
    User::LeaveIfError(phone.Open(server, info.iName));

    RMobilePhone::TMobilePhoneLocationAreaV1 areaInfo;
    RMobilePhone::TMobilePhoneNetworkInfoV1 networkInfo;
    RMobilePhone::TMobilePhoneNetworkInfoV1Pckg networkIn-
    foPckg(networkInfo);

    TRequestStatus req;
    phone.GetCurrentNetwork(status, networkInfoPckg, areaInfo);
    User::WaitForRequest(status);

    cellID = areaInfo.iCellId;
    LAC = areaInfo.iLocationAreaCode;

    phone.Close();
    server.UnloadPhoneModule(_L("Phonetsy.tsy"));
    server.Close();

}
```

Table A.2: Code for obtaining current Cell ID and Location Area Code on devices supporting the UIQ 2.x SDK

Appendix B

GSM Network Characteristics

The following sections list the observed characteristics of the GSM networks surveyed in Section 4.1, along with each network topology as visited in the traces. The full trace files are omitted for brevity but are available from the author on request[1].

B.1 Telstra GSM Network, Sydney, Australia

B.1.1 Network Parameters

No. Cells	437
No. LAs	9
Sub-traces	25
Total Duration	3395851 s
Cell Crossings	11684
Mean Movement Rate	12.39 moves/hour
Location Updates	566
LU rate	0.600 updates/hour
Mean Cell Dwell Time	290.64 s
Mean LA Residence Time	5999.74 s

Table B.1: Observed network trace parameters for Telstra Sydney GSM Network

[1] The author, James Cowling, may be contacted by email at jcowling@it.usyd.edu.au or via the Advanced Networks Research Group, School of IT, The University of Sydney, Australia.

B.1.2 Observed Topology

Figure B.1: Partial graph abstraction of Telstra Sydney GSM Network

B.2 Telstra GSM Network, Melbourne, Australia

B.2.1 Network Parameters

No. Cells	109
No. LAs	5
Sub-traces	1
Total Duration	253969 s
Cell Crossings	1190
Mean Movement Rate	16.87 moves/hour
Location Updates	223
LU rate	3.161 updates/hour
Mean Cell Dwell Time	213.42 s
Mean LA Residence Time	1138.87 s

Table B.2: Observed network trace parameters for Telstra Melbourne GSM Network

B.2.2 Observed Topology

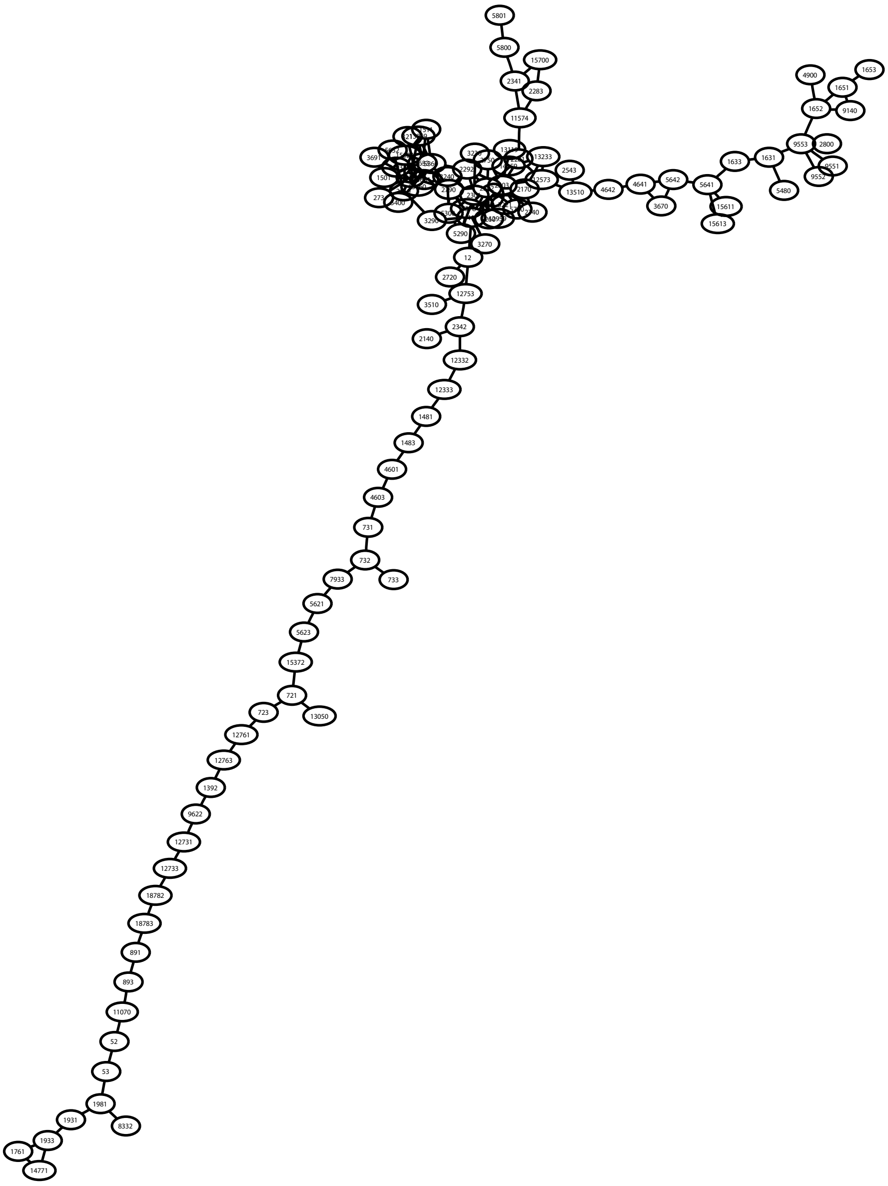


Figure B.2: Partial graph abstraction of Telstra Melbourne GSM Network

B.3 Orange GSM Network, London, UK

B.3.1 Network Parameters

No. Cells	276
No. LAs	18
Sub-traces	1
Total Duration	198836 s
Cell Crossings	1582
Mean Movement Rate	28.64 moves/hour
Location Updates	74
LU rate	1.340 updates/hour
Mean Cell Dwell Time	135.67 s
Mean LA Residence Time	2686.97 s

Table B.3: Observed network trace parameters for Orange London GSM Network

B.3.2 Observed Topology

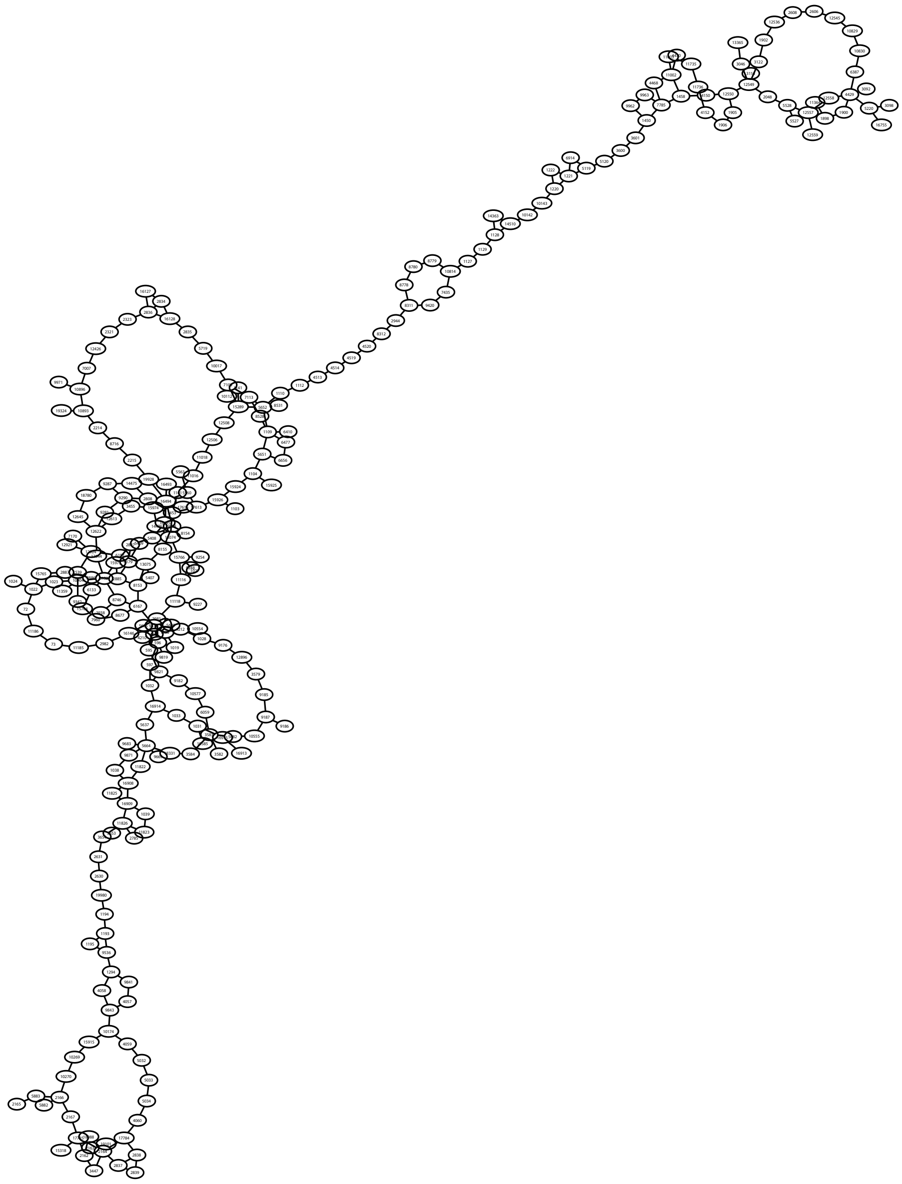


Figure B.3: Partial graph abstraction of Orange London GSM Network

B.4 Orange GSM Network, Paris, France

B.4.1 Network Parameters

No. Cells	345
No. LAs	15
Sub-traces	3
Total Duration	294074 s
Cell Crossings	1734
Mean Movement Rate	21.23 moves/hour
Location Updates	124
LU rate	1.518 updates/hour
Mean Cell Dwell Time	169.59 s
Mean LA Residence Time	2371.56 s

Table B.4: Observed network trace parameters for Orange Paris GSM Network

B.4.2 Observed Topology

Figure B.4: Partial graph abstraction of Orange Paris GSM Network

B.5 Orange GSM Network, Hong Kong, Hong Kong

B.5.1 Network Parameters

No. Cells	220
No. LAs	7
Sub-traces	2
Total Duration	195442 s
Cell Crossings	1285
Mean Movement Rate	23.67 moves/hour
Location Updates	44
LU rate	0.810 updates/hour
Mean Cell Dwell Time	152.10 s
Mean LA Residence Time	4441.86 s

Table B.5: Observed network trace parameters for Orange Hong Kong GSM Network

B.5.2 Observed Topology

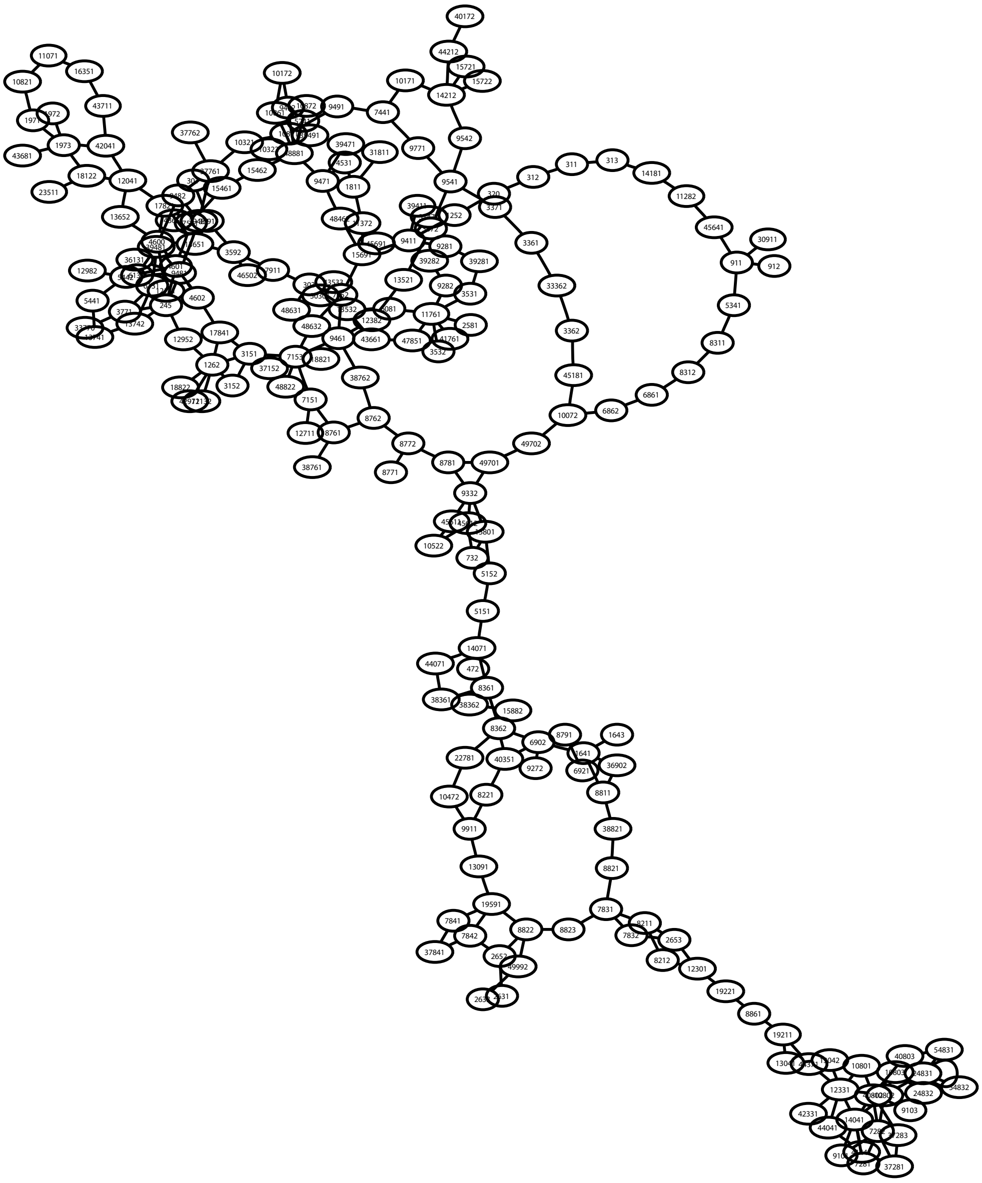


Figure B.5: Partial graph abstraction of Orange Hong Kong GSM Network

B.6 Vodafone GSM Network, Rome, Italy

B.6.1 Network Parameters

No. Cells	218
No. LAs	5
Sub-traces	2
Total Duration	199826 s
Cell Crossings	1359
Mean Movement Rate	24.48 moves/hour
Location Updates	58
LU rate	1.0450 updates/hour
Mean Cell Dwell Time	147.04 s
Mean LA Residence Time	2445.28 s

Table B.6: Observed network trace parameters for Vodafone Rome GSM Network

B.6.2 Observed Topology

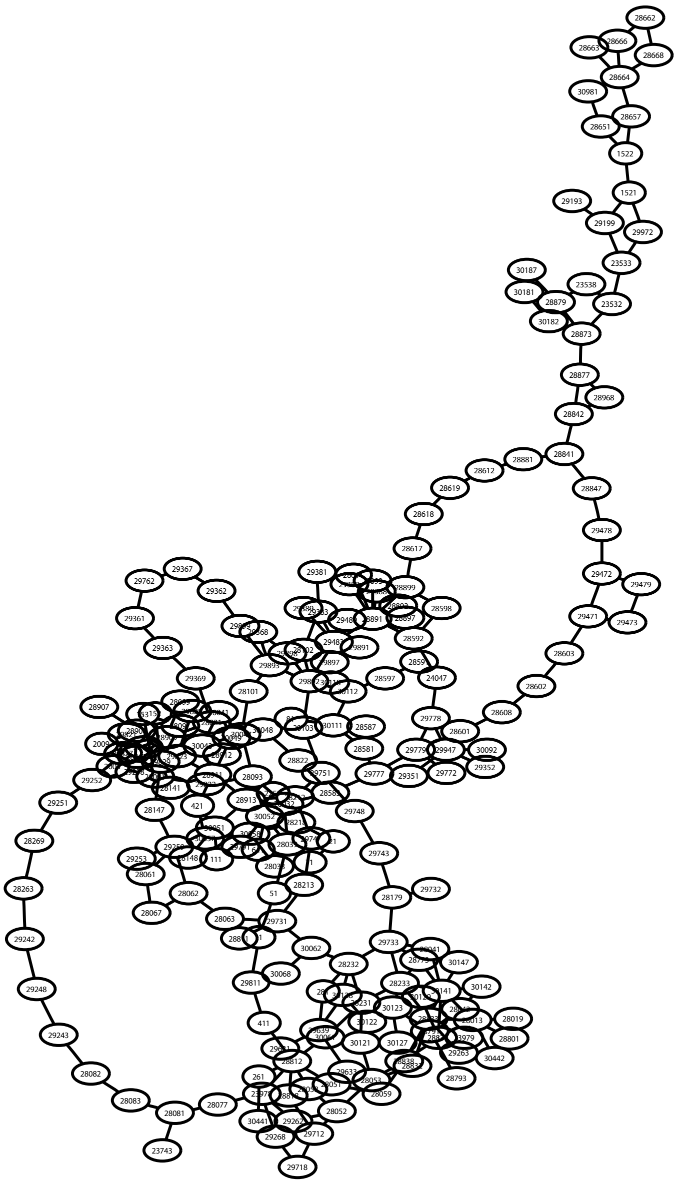


Figure B.6: Partial graph abstraction of Vodafone Rome GSM Network

Appendix C

Cell ID Hashes for Bloom Filter Example

The following are sample hash vales for the cells used in Section 6.4.5, used as input to the Bloom filter. They are uniformly and randomly distributed over all indices in the filter to assure minimum likelihood of false positives.

C.1 LA Cell Hashes

Cell ID	Hash 1	Hash 2	Hash 3
1311	75	90	95
1312	70	29	77
1323	4	47	93
6643	7	31	19
7421	82	25	0
7422	55	79	51
7423	25	28	95
7771	71	75	49
8051	54	65	28
9501	84	58	28
9702	82	40	14
9752	8	66	76
10672	74	60	83
10771	42	88	35
10781	29	48	66
10783	76	62	49
14162	95	29	54
24951	33	72	63
24952	4	44	23
24955	85	70	28
25151	66	78	32
25162	83	43	19
25165	70	51	8

Table C.1: Hash values for each cell in sample LA of Section 6.4.5

C.2 Neighbour Cell Hashes

Cell ID	Hash 1	Hash 2	Hash 3
1321	46	68	3
1322	71	13	16
1331	37	64	73
1333	40	45	45
1661	66	56	27
6611	34	26	83
5463	82	94	41
7603	45	95	83
7773	66	7	16
8052	61	9	79
8053	18	90	43
8451	60	95	87
9502	44	48	93
9503	46	56	63
10702	49	95	16
10721	53	65	80
10773	1	40	65
14271	20	10	63
16370	36	37	15
23941	94	97	71
23942	48	4	21
23943	52	67	31

Table C.2: Hash values for each neighbour cell of sample LA in Section 6.4.5

Declaration

The work in this thesis is based on research carried out as part of the qualifications for a bachelor degree with honours in Computer Science and Technology (Advanced) at the School of Information Technologies, The University of Sydney. No part of this thesis has been submitted elsewhere for any other degree or qualification and is original work unless referenced to the contrary in the text.