Technical Report

UCAM-CL-TR-741
ISSN 1476-2986

Number 741

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Vehicular wireless communication

David N. Cottingham

January 2009

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted September 2008 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Churchill College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

Vehicular wireless communication
David N. Cottingham

Transportation is vital in everyday life. As a consequence, vehicles are increasingly equipped with onboard computing devices. Moreover, the demand for connectivity to vehicles is growing rapidly, both from business and consumers. Meanwhile, the number of wireless networks available in an average city in the developed world is several thousand. Whilst this theoretically provides near-ubiquitous coverage, the technology type is not homogeneous.

This dissertation discusses how the diversity in communication systems can be best used by vehicles. Focussing on road vehicles, it first details the technologies available, the difficulties inherent in the vehicular environment, and how intelligent handover algorithms could enable seamless connectivity. In particular, it identifies the need for a model of the coverage of wireless networks.

In order to construct such a model, the use of vehicular sensor networks is proposed. The Sentient Van, a platform for vehicular sensing, is introduced, and details are given of experiments carried out concerning the performance of IEEE 802.11x, specifically for vehicles. Using the Sentient Van, a corpus of 10 million signal strength readings was collected over three years. This data, and further traces, are used in the remainder of the work described, thus distinguishing it in using entirely real world data.

Algorithms are adapted from the field of 2-D shape simplification to the problem of processing thousands of signal strength readings. By applying these to the data collected, coverage maps are generated that contain extents. These represent how coverage varies between two locations on a given road. The algorithms are first proven fit for purpose using synthetic data, before being evaluated for accuracy of representation and compactness of output using real data.

The problem of how to select the optimal network to connect to is then addressed. The coverage map representation is converted into a multi-planar graph, where the coverage of *all* available wireless networks is included. This novel representation also includes the ability to hand over between networks, and the penalties so incurred. This allows the benefits of connecting to a given network to be traded off with the cost of handing over to it.

In order to use the multi-planar graph, shortest path routing is used. The theory underpinning multi-criteria routing is overviewed, and a family of routing metrics developed. These generate efficient solutions to the problem of calculating the sequence of networks that should be connected to over a given geographical route. The system is evaluated using real traces, finding that in 75% of the test cases proactive routing algorithms provide better QoS than a reactive algorithm. Moreover, the system can also be run to generate geographical routes that are QoS-aware.

This dissertation concludes by examining how coverage mapping can be applied to other types of data, and avenues for future research are proposed.

In memory of my grandfathers

Rowland D. Cottingham
(1916–2006)

Suraj B. Mehra
(1918–1972)

Contents

Figures

Tables

Acknowledgements

"Of making many books there is no end,
and much study wearies the body."

— Ecclesiastes 12:12b (NIV)

I am indebted to the following, without whom this work would not have been possible:

- Jesus Christ, for his mercy and many blessings.
- My wife, Elke, my parents, Peter and Jackie, my sister, Ruth, my parents-in-law, Albert and Lutgart, and my brother-in-law Guy for their love and support over the years.
- Andy Hopper for his supervision and encouragement throughout.
- Robert Harle for his frequent helpful input and advice, and inciteful comments on this manuscript.
- Jonathan Davies, Andrew Rice, and Tom Craig for their patience and encouragement as office mates.
- Ian Wassell, Brian Jones, Alastair Beresford and many others in the Digital Technology Group, as well as Jon Crowcroft in the Systems Research Group and Glenford Mapp at Middlesex University for allowing me to pick their brains.
- Dina Papagiannaki and Adrian Stephens formerly at Intel Research Cambridge, for their advice on various aspects of the IEEE 802.11 standard, and loan of equipment.
- Joseph Newman, Andrew Rice, Jonathan Davies, Tim Griffin and Alan Jones for in depth discussion on the applications of routing metrics, and Simon Hay and Ioannis Chatzigeorgiou for their helpful comments on multi-planar routing.

- The staff at the University of Cambridge Computer Laboratory for the thousand "little" things that were needed.
- The Computer Laboratory for its generous financial support.
- My examiners, Jon Crowcroft and Kyle Jamieson, for their helpful comments that increased the manuscript's clarity.

Parts of this work were carried out in conjunction with other people. In particular:

- The Sentient Van platform was joint work with Jonathan Davies and Brian Jones, who performed the majority of the hardware and software deployment. The author's involvement was in deploying communications infrastructure, adding cameras, and in general maintenance. Further details of the platform are given in [53, 40].
- The original idea for representing coverage as a multi-planar graph was conceived by Richard Gibbens. All further concepts, implementation details, and routing theory are the author's own work.

[1] `http://creativecommons.org/licenses/by-sa/2.0/`

Publications

The following have been entirely incorporated into this dissertation:

David N. Cottingham and Robert K. Harle. Constructing Accurate, Space- efficient, Wireless Coverage Maps for Vehicular Contexts. *Proceedings of the 4th International Wireless Internet Conference (ICST WICON)*, November 2008, Hawaii, USA.[41]

David N. Cottingham and Robert K. Harle. Handover-optimised Routing Over Multi-planar Graphs for Vehicles. *In progress*, 2009.[42]

David N. Cottingham, Ian J. Wassell, and Robert K. Harle. Performance of IEEE 802.11a in vehicular contexts. In *Proceedings of the IEEE Vechicular Technology Conference*, pages 854–858, April 2007, Dublin, Ireland.[44]

In addition, some content has been taken from, or is similar to the following:

Jonathan J. Davies, David N. Cottingham, and Brian D. Jones. A Sensor Platform for Sentient Transportation Research. In *Proceedings of the 1st European Conference on Smart Sensing and Context*, Lecture Notes in Computer Science volume 4272, pages 226–229, October 2006, Enschede, The Netherlands.[53]

David N. Cottingham, Jonathan J. Davies, and Brian D. Jones. A Research Platform for Sentient Transport. *IEEE Pervasive Computing*, 5(4):63–64, Oct–Dec 2006.[40]

David N. Cottingham and Jonathan J. Davies. A Vision for Wireless Access on the Road Network. In *Proceedings of the 4th International Workshop on Intelligent Transportation*, pages 35–30, March 2007, Hamburg, Germany.[39]

David N. Cottingham and Pablo Vidales. Is Latency the Real Enemy in Next Generation Networks? In *Proceedings of the 1st International Workshop on Convergence of Heterogeneous Wireless Networks (ICST ConWiN)*, July 2005, Budapest, Hungary.[43]

Finally, publications arising from other work the author has carried out or been involved with are:

David N. Cottingham, Alastair R. Beresford, and Robert K. Harle. A Survey of Technologies for the Implementation of National-scale Road User Charging. *Transport Reviews*, 27(4):499–523, July 2007.[37]

David N. Cottingham, Jonathan J. Davies, and Alastair R. Beresford. Congestion-aware Vehicular Traffic Routing Using WiFi Hotspots. In *Proceedings of Communications Innovation Institute Workshop*, pages 4–6. Cambridge-MIT Institute, April 2005, Cambridge, UK.[38]

Pablo Vidales, Carlos J. Bernardos, Ignacio Soto, David Cottingham, Javier Baliosian, and Jon Crowcroft. MIPv6 Experimental Evaluation Using Overlay Networks. *Computer Networks*, 51(10):2892–2915, July 2007.[234]

Glenford Mapp, David N. Cottingham, Fatema Shaikh, Pablo Vidales, Leo Patanapongpibul, Javier Balioisian, and Jon Crowcroft. An Architectural Framework for Heterogeneous Networking. In *Proceedings of the 1st International Conference on Wireless Information Networks and Systems (WINSYS)*, August 2006, Setubal, Portugal.[158]

Jon Crowcroft, David Cottingham, Glenford Mapp, and Fatema Shaikh. Y- Comm: A Global Architecture for Heterogeneous Networking. In *Proceedings of the 3rd Annual International Wireless Internet Conference (WICON)*, October 2007, Paris, France. Invited paper.[47]

Glenford Mapp, David Cottingham, Fatema Shaikh, Edson Moreira, Renata Vanni, Wayne Butcher, Aisha El-safty, and Jon Crowcroft. An Architectural Framework and Enabling Technologies for Heterogeneous Networking. Submitted to *Journal of IEEE/ACM Transactions on Networking*, 2008.[157]

Edson D. S. Moreira, David N. Cottingham, Jon Crowcroft, Pan Hui, Glenford E. Mapp, and Renata M. P. Vanni. Exploiting Contextual Handover Information for Versatile Services in NGN Environments. In *Proceedings of the 2nd IEEE International Conference on Digital Information Management (ICDIM)*, volume 1, pages 506–512, October 2007, Lyon, France.[165]

Bogdan Roman, Frank Stajano, Ian Wassell, and David N. Cottingham. Multi-carrier Burst Contention (MCBC): Scalable Medium Access Control for Wireless Networks. In *Proceedings of the IEEE Wireless Communications & Networking Conference (WCNC)*, pages 1667–1672, March 2008.[193]

1

Introduction

TRANSPORT IS indispensable in modern day life, both for business and private users. Whilst in an ideal world society's appetite for travel would be less, the fact remains that it is growing. In order to make transport safer, cheaper, and more efficient, manufacturers are increasingly turning to computing technologies for help. Vehicles are becoming sensor-rich platforms, able to react to changes in their surroundings. As a consequence, wireless communications infrastructure has a vital rôle to play in enabling information to reach vehicles, and vehicles to upload data concerning their environment.

The enormous diversity of wireless networks is a double-edged sword. A great benefit is the availability of connectivity, in some form, practically anywhere on the globe. However, such heterogeneity in both number of technologies and number of networks of each technology means it is difficult to correctly choose which to use. Moreover, each time a user changes the network they connect to, they incur a penalty in the form of a disconnection (or disruption) for multiple seconds. Changing network every few seconds is therefore not a viable proposition. Unfortunately, vehicles move at high speeds, and hence move in and out of the coverage areas of wireless networks quickly. Without a system that intelligently selects the network to connect to, vehicular connectivity will be suboptimal.

This dissertation proposes and evaluates mechanisms to enable such intelligent network selection for vehicles, thus taking advantage of network diversity to provide better network QoS.

1.1 The Evolution of Transport

The last decade has seen radical changes in both how society views Internet connectivity and transportation. At a cursory glance many people would not imagine the intersection of these two fields as being particularly large; after all, Internet connectivity allows consumers to obtain information or access online services, whilst transport involves the movement of people and goods. Why, therefore, is research necessary at this intersection?

1.1.1 Transport as a Right

We begin by examining how our views of transport have changed. Man has changed his mode of transport radically over time, first by harnessing other sources of energy (animal, or later, combustible fuel) to avoid expending his own physical energy, and latterly by devolving responsibility for the control of transport to avoid the drain on mental resources. We can chart this evolution through many centuries of history: palanquins carried by slaves, succeeded by horses or horse-drawn coaches, the dawn of the motor car (possibly with an attendant chauffeur), and the arising of (mass) public transportation. The ability for humans to migrate long distances with little physical effort has become a right enshrined in the public psyche, rather than a privilege that perhaps very few could afford. With this increased expectation for mobility has come the desire for privilege in another form, namely in the reduction of cognitive load associated with transportation. Whereas previously social status was dictated by the ability to travel, it is now by the ability to do so with as much of the burden removed from the traveller as possible, as well as with as much individual comfort as practical.

Such changes are borne out by the range of transport options available to us: cars are now common place in western society, with many households having two or more, and public transport outside of major cities being underutilised. A limited number of car owners also have chauffeurs. However, the price differential between a self-drive vehicle and a chauffeured one (that is still private, rather than mass transit) is significant. It was into this niche, between self-drive and chauffeur, that the insertion of another price differentiator was necessary in order to continue vehicle-manufacturers' trend of profit expansion. Technology is that differentiator, allowing the distinction in price to be made between the basic vehicle seen as a "right", and more luxury models.

1.1.2 Transport as a Killer

Concurrent with the rise in transport usage, deaths and injuries caused by vehicles also increased. Compounded with more effective health care that decreased mortality arising from disease, governments have moved their focus from making transport available to all to making it safer. Figure 1.1 shows how in recent years the trend in UK road fatalities has been a downward one, despite the number of miles travelled by road vehicles increasing. Clearly statutory instruments, such as the requirement for seat belts to be fitted to all vehicles, have increased safety.

However, governments are now observing that the number of deaths and serious injuries due to road transport per year is beginning to plateau. This is acting as an incentive to find novel technologies to continue the downward trend. With costs of road building spiralling, building more capacity to decrease congestion is not seen as a long-term solution. However, accidents are significantly more likely on congested motorways as compared to free-flow conditions [20]. Meanwhile,

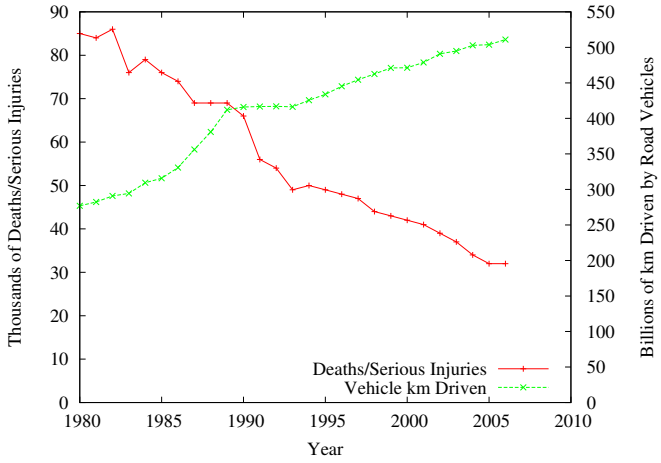


Figure 1.1: Deaths and serious injuries occurring, and miles driven by road vehicles, on UK roads since 1980 [59].

passive safety features that can be engineered into vehicle designs, such as crumple zones or air bags[1] are now only subject to small incremental improvements, rather than significant leaps in efficacy. Other, more advanced, solutions must therefore be found.

1.2 The Evolution of Connectivity

1.2.1 Complexity: a Product of Diversity

The changes in transport that were briefly charted in the previous section took place over a considerable period of time. In contrast, the Internet has grown from a research project to being indispensible within only two decades. Connectivity has evolved from low throughput, high latency, high cost channels such as human messengers carrying paper (and hence available only to the rich), to third generation cellular telephones, capable of multimedia transmission and available to the vast majority of the public[2]. Demand for data services is predicted to outstrip that for voice communications, highlighting how in a comparatively short space of time machine-to-machine communication has become a crucial element of our daily lives.

[1] Technically, air bags are active in that they are activated by sensors rather than the driver. However, the technology involved is relatively simple, and does not require large computational resources.

[2] It is interesting to note that *mobility* in communications is still inferior to when messengers were used: we still find areas where our cellular handsets have no coverage. What has increased is the *range* over which communications are possible, messengers not being suitable for intercontinental communications in quite the same way as the cellular Short Message Service.

Due to the diverse nature of the technologies available, consumer devices now regularly come with transceivers for two or more of these technologies built-in. Until recently each interface was regarded as being useful for particular applications, such as the cellular interface for voice calls, Bluetooth for personal area networks, and high data rate (but not everywhere-used) applications taking advantage of an 802.11b/g WiFi interface where a hotspot was available. Today, operators are beginning to develop techniques for utilising whichever interface is best suited to the task, such as switching to Voice over IP (VoIP) using the WiFi interface when the device is in the vicinity of the owner's home[3]. Although such handover operations sound simple, complications arise both in deciding when handovers should take place, and in making such handovers seamless.

1.2.2 Quantifying Network Diversity

The sheer number and density of network deployments in a typical city brings an enormous diversity in both technology and ownership. This is made even more complex, as networks not only have small coverage areas, but overlap in dense deployments.

Figure 1.2 illustrates the sheer number of different cellular network base stations available in a 9 square km area of each of the cities of London and Cambridge, UK. Whilst at present the majority of users use only one provider's infrastructure, there is still likely to be a choice (normally made by the network infrastructure, rather than the cellular handset) of base stations to connect to in many areas.

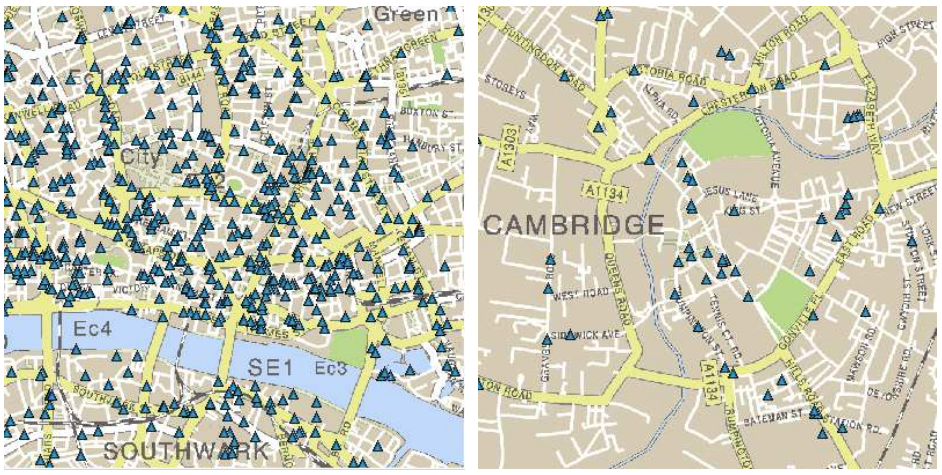
Meanwhile, data collected by the Sentient Van (Section 3.1) reveals that in the city of Cambridge alone, over 3800 distinct 802.11b/g wireless networks have been recorded, with many roads in the city not visited by the vehicle. In Cambridge, USA, the CarTel project reported that they recorded over 32,000 distinct networks over one month, with sensors deployed on nine cars [24]. These networks have very small coverage areas, but are densely deployed, with a significant degree of overlap. In some cities in the USA, an overlap of each AP with three others was very common, with some APs overlapping with up to 85 others [3].

Such large numbers of networks mean that choosing which network to connect to at any one time is complex. Without information concerning the coverage and likely performance of each network, such a choice is ill informed, and hence the sequence of networks that a user connects to is likely to be very suboptimal.

Therefore, whilst contributing to making communication ubiquitous, untethered technology has also brought into sharp relief how localised many forms of such communication are. Users of the GSM cellular network find few geographical locations that are not serviced by at least one operator, and hence voice calls can

[3] See, for example, BT's Fusion product, `www.bt.com/btfusion/`.

[3] `http://www.sitefinder.ofcom.org.uk/`



(a) London: 626 locations, 192 shared

(b) Cambridge: 65 locations, 21 shared

Figure 1.2: Cellular base stations in the cities of London and Cambridge, UK. Base stations are shown by blue triangles, with some hosting more than one operator or technology (shared). Diagrams taken from Ofcom's SiteFinder[3], used with permission. Copyright 2008 Ofcom.

be made from practically anywhere. However, as technology advances and more complex forms of transmission become achievable, transmission ranges decrease and costly deployments are limited to those areas that will yield a good rate of economic return. WiFi hotspots, whilst offering high throughputs, are limited to approximately 200 metres in range, and are generally only deployed in centres of population. Thus, connectivity has evolved to be provided by a rich mix of technologies, each having its own applications, rather than there being one universal system.

1.3 Sentient Transportation

Intelligent Transportation Systems, ITS, stem from the application of computing and communication technologies to the field of transport. ITS is a nascent field, having arisen only recently from the dual demands for further increases in passenger comfort and the needs of governments to increase safety and yet manage demand. Whilst manufacturers have thus far tended to concentrate on the deployment of greater computing resources on vehicles to further these aims, it is only recently that research has examined how communications can play a significant rôle in ITS.

1.3.1 Extending Ubiquitous Computing to Vehicles

The deployment of computing infrastructure in vehicles is a manifestation of the concept of ubiquitous computing, a term coined by Mark Weiser to describe how computing resources would be present, and yet invisible, in all day-to-day objects [238]. Anti-lock braking systems on road vehicles utilise a microprocessor to detect when a skid is occurring, and vary brake pressure appropriately, yet drivers do not feel that they are "using a computer" every time they slow down. Such disappearance into the periphery of users' consciousnesses is precisely what ubiquitous computing aims to achieve. This is in marked contrast to how standard workstations are used, where for many people a great deal of cognitive load is implied.

Ubiquitous computing not only implies humans interacting with hundreds of computers each day, but also depends on those computers *communicating* with each other, in order to share the information each is provided with. Sentient computing builds on widely deployed sensing and computing infrastructure in using it to make decisions and carry out actions that are context-aware, and hence intuitively correct to users [109]. When applied to transport, we term this *Sentient Transportation*. A very simple vehicular example is using the location of a vehicle provided by a GPS receiver, and a digital road map, to infer that the vehicle is approaching a tunnel and that the headlights should therefore be switched on. Such intuitive decisions can only be taken if all the computing infrastructure is interconnected. To illustrate this, the different communications paths in this small example are listed below:

- Signals from satellites orbiting the earth are received by the GPS unit.
- The resulting location fix is queried over the vehicle's internal network by the navigation system.
- Digital maps used by the navigation unit are updated using a cellular link to ensure they include the latest details concerning the road (e.g. roadworks, traffic conditions, areas where there may be ice on the tarmac).
- A message is sent over the internal network to the vehicle's management computer, which infers what actions need to be performed.
- Again using the vehicle's internal network, a message is sent to the actuator for the headlights, causing them to illuminate.

This simple example illustrates how location and communication systems are central to sentient computing. Of course, myriad other, more complex, sentient transportation applications are possible [4], and indeed are already being deployed. Asset tracking on vehicles, intersection collision avoidance (where inter-vehicular communication occurs), and platoon driving (many vehicles are closely packed on

a motorway, accelerating and braking in a co-ordinated fashion), are all concepts that are set to become reality in the medium term. All rely heavily on communications infrastructure being integrated into vehicles and their environment.

Whilst inter-vehicular communications are envisaged as being central to the enhancement of safety, most commercial benefit is likely to come from vehicle to infrastructure communication. This takes place between vehicles and fixed base stations, which are already deployed in a variety of forms. This dissertation concentrates on how vehicle-to-infrastructure communications may be improved, chiefly because there is no single technology that is used, or indeed that is suitable, for all of the many different vehicle-to-infrastructure applications. In addition, this work further specialises by considering those applications that require continuous (or near-continuous) connectivity, rather than solely sporadic network access (such as might be provided by "plugging in" a vehicle overnight when parked in its home garage). Examples of these applications include:

- Asset tracking
- On the move Internet access, particularly for public transport
- Emergency or fleet vehicle information download/dispatch
- Mobile working, especially for the construction industry or other outdoor work
- Voice/Video over IP conversations
- Utilising vehicles for large scale real-time sensing
- Semi-real time analysis of vehicle operating data for remote diagnosis of problems.

This type of application generally necessitates as few network disconnections or disruptions (such as packet losses or drops in throughput) as possible. As network deployments become more complex, and applications' demands grow, such disruption will be ever more difficult to avoid.

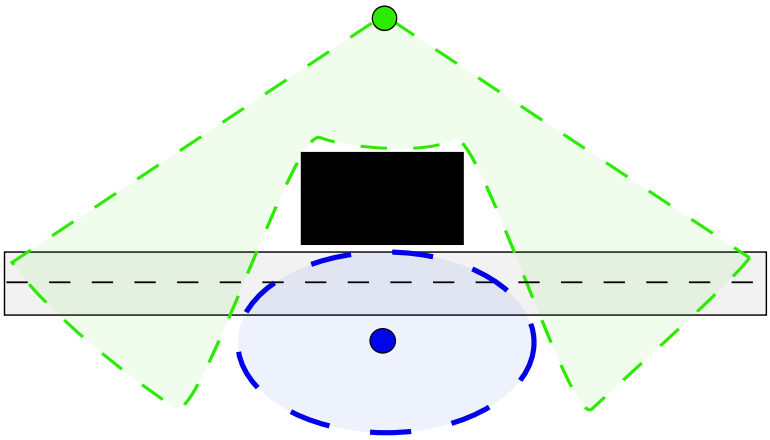


Figure 1.3: An illustration of why a world model for communications networks is needed (see text).

1.3.2 Communication Pitfalls

As outlined in Section 1.2, whilst there are an enormous number of different wireless networks deployed today, this does not mean that users experience ubiquitous connectivity of uniformly good quality. Instead, we find that as the diversity of networks increases, the complexity involved in most efficiently utilising them raises significant issues. Moreover, when we consider connectivity for vehicles, we find that issues such as speed of movement exacerbate the problems experienced by variation in coverage with geography. These problems are illustrated by the following scenarios:

Lack of a world model: Figure 1.3 depicts a typical road, covered by two different wireless networks. One, coloured green in the diagram, does not cover a small portion of the road due to the presence of a building. Peter, a lorry driver, travels along the road. An algorithm that is used onboard his vehicle to select which network to connect to will first connect to the green network. If it has no model of what the coverage of the networks available is, when the green network becomes unavailable, it will handover to the blue network, causing a disconnection period of 4 seconds. Unfortunately, the time Peter spends in the blue coverage area is very short, and no sooner is the handover complete but the network becomes unavailable. A handover to the green network (now available once more) takes place, causing Peter's contact with his dispatcher to be interrupted once more for 5 seconds.

Environment not well understood: Ruth has been told by her wireless network hotspot provider that connectivity is available within 100 metres of any of their hotspots. Ruth tries to connect to one, whilst waiting in her car, just 50

metres down the street from the cafe where the hotspot is located. She finds she can't connect. As she is moving her car further away from the cafe, she hears an alert from her instant messenger application: she's now connected. Confused, Ruth decides not to bother using the hotspot next time.

Communications not a specifiable requirement: as the requirement for connectivity from vehicles metamorphoses from best-effort to guaranteed quality of service (QoS), users will need to be able to express their preference of how they trade-off connectivity quality with journey length, or economic cost. Jackie is travelling to a conference to give a presentation. She is late, and therefore does not wait for the presentation file to finish downloading from the corporate network before leaving the office. Jackie's laptop finishes the transfer over the (expensive) cellular network. Had she driven through a side street, she would have passed by a WiFi hotspot belonging to a scheme that her company subscribes to, and the transfer would have been completed more cheaply and in less time. Unfortunately, there is no way for her to tell her navigation unit to take connectivity, as well as length of route, into account.

1.4 Understanding, Providing, Modelling, and Optimising Wireless Communications for Vehicles

Current understanding and approaches to providing communications capabilities for vehicles are fragmented, with little work on how the multiplicity of technologies available can be used in concert in order to achieve best performance. The work detailed in this dissertation concerns how mobile sensor platforms may be used to gather data about their environment; what algorithms can be used to process the large quantities of data collected; and how the results may then be used for multi-criteria routing to improve network connectivity. Specifically, the contributions made in this work address the following questions:

What performance can be expected for wireless technologies in vehicular environments? Vehicles travel at high speeds and over far larger areas than indoor users. The goal here is to investigate how these environmental factors affect off-the-shelf wireless technologies, and how these technologies are affected by other factors such as weather, time of day, or position.

How can an accurate, compact, model of the wireless environment be constructed? If large quantities of sensor data concerning the wireless environment that vehicles operate in are available, what algorithms can be used to process this data efficiently? How accurate compared to the real world are the data representations produced?

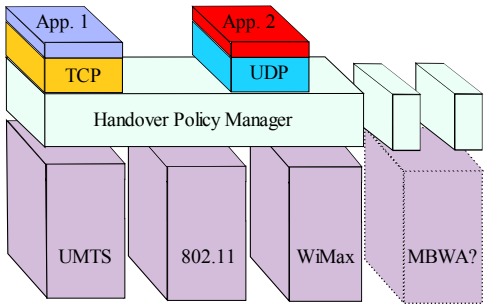


Figure 1.4: How a proactive handover layer that utilises extrinsic information relates to the different technology stacks in use. The handover layer chooses the best stack to use at any one time, whilst connectivity is seamless for the transport and application layers.

How can such models be used, particularly for optimising communications systems? Here, the goal is to use the processed sensor data in order to enhance decision-making by vehicles' onboard systems. In particular, how can vehicles' connectivity requirements be taken into account when performing routing? How can such requirements be formally expressed as multi-criteria routing problems? Any solution must not require a minimum installed base of users, and ideally should not require any changes to existing infrastructure.

1.5 Limitations of Scope

Providing connectivity to moving vehicles is an extremely broad research area, with significant work carried out at all levels of the protocol stack. The work described in this dissertation seeks to take advantage of the benefits of each network technology stack by creating a layer that is aware of extrinsic information concerning the coverage and performance of each technology, as shown in Figure 1.4. Therefore, whilst each stack may itself present opportunities for optimisation, that is not the focus of this work. Concretely, this dissertation limits its scope as follows:

Optimising the use, rather than the technology. This dissertation aims to propose mechanisms by which current or emerging wireless technology deployments can be best used. The communications engineering aspects of individual technologies, such as the antennas used or the mathematical aspects of modulation or coding schemes, are not examined here. There exists a large body of work on such areas, both for vehicular and non-vehicular applications. In contrast, this dissertation shows that none of the available technologies constitutes a "silver bullet" on their own, and hence concen-

trates on how to take advantage of the diversity of wireless technologies to enhance the connectivity provided to vehicles.

Technology selection rather than protocol optimisation. This dissertation is not concerned with optimising higher layer transport protocols such as TCP. Such protocols have been extensively researched, and their flaws when used over wireless communication links are well known [251]. Whilst modifications to TCP have been proposed, they require widespread deployment in both clients and servers in order to be of use. In contrast, by performing network selection more wisely, transport protocol performance can be increased.

Land-based vehicles only. This dissertation focuses on land-based vehicles, given that these are greatest in number, travel on very constrained routes, and the most easily available for evaluation purposes. However, the majority of the algorithms proposed in this work could also be applied to air- and sea-based transportation, provided that the movements took place along fixed routes. This is not an unreasonable constraint, given the existence of air and shipping corridors.

1.6 Dissertation Outline

The structure of the remainder of this dissertation is as follows:

Chapter 2 introduces the concept of Intelligent Transportation Systems, overviews the technologies currently used for vehicular communication, explains why handovers between them are difficult, and how wireless coverage maps can help.

Chapter 3 examines two technologies used for communication with vehicles, UMTS and IEEE 802.11, in terms of their performance in the vehicular environment. The Sentient Vehicles project is introduced as a platform for carrying out these experiments. In addition, data are provided concerning the effects of meteorology and geography on these technologies.

Chapter 4 presents the concept of coverage mapping as a solution to the problem of network selection complexity. Novel algorithms are proposed to process large quantities of signal strength data into coverage maps. The results are then evaluated using further real traces, to assess their accuracy.

Chapter 5 demonstrates how coverage maps can be converted into a multi-planar directed graph, which is able to express the costs of handing over between different networks. The complexity of the graph is analysed, and techniques for reducing it presented.

Chapter 6 overviews and develops graph theory for multi-criteria routing, and suggests metrics to use for shortest path routing over the multi-planar graphs generated previously. These pro-active handover decision metrics are then evaluated against a realistic reactive handover decision algorithm.

Chapter 7 concludes with a summary of the contributions made in this dissertation, comparing them to the goals cited above, and overviews avenues for further research.

2

Background

THIS Chapter overviews the broad area within which this dissertation falls. Beginning with a description of Intelligent Transportation Systems (ITS), it then moves on to describe how ITS has given rise to vehicular sensor networks. The network technology choices for such networks are then overviewed, before examining in detail how two important communication technologies used for ITS, namely UMTS and IEEE 802.11x, function. The focus then shifts to detailing how utilising the multitude of different communication systems available is difficult, mainly due to the disconnection times incurred when handovers between different technologies take place. Such difficulties can be mitigated by means of intelligent, proactive handover algorithms, including those based on wireless network coverage maps. Finally, an overview of how coverage maps for wireless positioning systems are currently constructed is presented, and an explanation given for why these are unsuitable for vehicular applications.

2.1 Intelligent Transportation

Intelligent Transportation Systems (ITS) can be defined as the application of computing equipment and algorithms to enhance any method of transportation. Such enhancements may provide increased safety, lowered cost (economic, temporal, environmental), increased comfort, or have a greater efficiency. A wide variety of systems are counted under this umbrella, a few examples being electronic ticketing systems, anti-lock braking, traveller information systems, and congestion charging. In this dissertation, the focus is confined to those systems that concern road vehicles, and, moreover, involve the deployment of computational and communications infrastructure on such vehicles. For an overview of the general field of ITS, see chapters 3 and 4 of [35], or [18].

For road vehicles, two categories of system can be identified: intelligent *infrastructure* and intelligent *vehicles*, the classification depending on where the bulk

of processing takes place (as in all cases communication will take place between the infrastructure and the vehicles). Examples[1] of these are given in Table 2.1.

[1] Further examples can be found at `http://www.itsoverview.its.dot.gov/`.

Intelligent Infrastructure	Intelligent Vehicles
• **Arterial Road Management:** variable speed limits, adaptive traffic light timings, variable message signs, ramp metering • **Incident Management:** automated incident detection, lane closures/direction reversals, hazardous cargo tracking • **Road tolling:** dynamic pricing, distance-based charging • **Maintenance:** at-base diagnosis of faults whilst on the move	• **Navigation Services:** route guidance, dynamic routing based on traffic conditions, location-based services • **Collision Avoidance:** lane-departure warning, pedestrian detection, stop-and-go cruise control, platoon driving • **Greater vision:** self-parking vehicles, infrared night-vision

Table 2.1: Example ITS Applications.

In order to bring about truly intelligent transportation, two ingredients are necessary. Firstly, vehicles must be equipped with sensors to gain information concerning their environment. Secondly, communications systems must be deployed in order to interact with other vehicles and fixed infrastructure. One early example was the California PATH project [103] which sought to address how vehicles could form platoons on major roads, communicating their velocity, acceleration, and position to the other members of the platoon. Clearly both the sensor and the communications technologies were crucial. Similarly, projects such as collision avoidance on roads, or allowing emergency vehicles priority at intersections without traffic lights, all require communications and sensor systems that can be relied upon. This need has driven the deployment of technology in vehicles, but in turn has given birth to a new set of both opportunities and challenges. Vehicular sensor networks are now a real possibility, but the problem of effectively utilising the wide variety of networks available has also arisen. This dissertation builds upon the ITS technologies currently being deployed to help solve the challenges of vehicular communications in the longer term.

2.2 Vehicular Sensor Networks

One growing application of intelligent transportation is the usage of large numbers of vehicles as mobile sensors. Applications range from inferring traffic speeds in real time [95] to the usage of GPS traces for updating digital road maps [52]. Vehicles within such deployments must not only possess sensing equipment, but

also have access to network connectivity in order to transfer the data, the complete system being known as *telematics* [229]. One of the best known projects in this field is General Motors' OnStar[2] programme, where drivers can request directions, remote unlocking (in the event of lost keys), or an equipped vehicle can autonomously contact the emergency services. Such services are the commercial force behind the deployment of sensors (such as GPS receivers) in vehicles. In turn, such sensors can be used for more complex ITS applications.

The concept of participatory sensing [22], where sensing infrastructure is owned by individual members of the public, holds much promise when applied to vehicles. The advantages of vehicular sensor networks (VSN) over traditional fixed sensors are several:

- **Positioned at ground level:** to avoid vandalism, fixed sensors tend to be positioned at the tops of poles. An example is pollution sensing, where the map provided by fixed sensors is of pollutant concentrations several metres up in the air. Vehicles are far better placed to measure quantities that may change significantly with altitude, at heights close to that of the average human being.

- **Dense coverage where it is most needed:** for many ITS-related applications, the most important areas for which sensor data is required are those where there are large numbers of vehicles (e.g. knowledge of congestion in city centres). In addition, this sensor density is not fixed, but instead moves with traffic density. A fixed deployment would need to provide a high density of sensors throughout a city, even if high traffic densities did not occur in all areas at all times. Here, the same density is available where and when it is needed.

- **Greater coverage for fewer sensors:** in many cases the sampling rate required to observe a phenomenon (e.g. pollutant concentration) is low. Hence, a fixed sensor will spend a significant fraction of time not being of use. With a sufficient number of mobile sensors, the sampling interval can be achieved, whilst collecting data concerning other locations during the otherwise wasted time.

- **Regular servicing:** vehicles are taken for servicing approximately once per year. This provides an ideal opportunity for sensor maintenance and testing, rather than the costly approach of sending a maintenance team to each fixed sensor site.

[2] `http://www.onstar.com/`

- **Power and network connectivity are readily available:** obtaining such resources for fixed sensors can be expensive, or if they are battery powered, energy constraints limit sensing intervals and communication distances. Vehicles have easily accessed power supplies, and in an increasingly ITS-equipped world, at least one network interface which has a range of hundreds of metres to kilometres.

Hence, it is important to investigate how to use these potentially large sensor networks to their full potential. In particular, how the communications aspect can be optimised to provide seamless connectivity.

2.2.1 VSN Deployments

The term *floating car data* [54] is used to describe the sensor readings garnered from moving vehicles. One example is the OPTIS project in Sweden [131], where 220 cars were equipped to report their speeds over cellular GPRS modems in real-time. This allowed the city to have a knowledge of congestion as good as their existing camera/loop-detector systems for comparatively low cost. An analogous project is taking place in Germany with city taxi fleets [221]. Similarly, another project equipped 200 cars to report their speeds to a central server every 30 seconds, with the aggregated data then being transmitted back to the vehicles. The onboard navigation units then used this information to provide updated route guidance to the driver [57]. The type of communications used varies: the SOTIS [242, 243], StreetSmart [65], and TrafficView [169] projects aim to distribute (and process) such data over a vehicular ad hoc network (Section 2.3.4) instead of a cellular network. Commercialisation of this type of data is underway: Inrix and Dash Navigation both use fleets of vehicles to provide real-time traffic information, whilst cellular network providers attempt to track large numbers of mobile phones along major roads in order to infer traffic speeds [5].

Floating car data is not only confined to speeds and positions. A wide variety of other sensors have been deployed, such as to infer the locations of potholes, or the stress on the driver at particular intersections. One well-known project is MIT's CarTel [112], where several cars were equipped with embedded computers, onboard diagnostics units for reading engine parameters, GPS receivers, and 802.11b/g wireless transceivers. The units recorded details of the wireless networks they encountered, and attempted to connect to the Internet through them, providing insight into the availability of WiFi hotspots for vehicles and the amount of data that can be transferred through them. The results showed that a median transfer of 216 KBytes per session was possible. Given that 32,000 unique networks were recorded over the experiment's duration [24], this suggests that such connectivity has great utility. Separately, the project also used accelerometers to record locations where the vehicles experienced motion that could be due to a pot-

hole in the road surface. Using further data processing techniques this enabled a map of pothole locations to be built up [73].

Another project of interest is BikeNet [71], where a bicycle was fitted out with a large number of sensors, including tilt, GPS position, speed, cyclist's heart rate and galvanic skin response, and pollutant and allergen sensors. Sensor data was uploaded to WiFi access points that the bike encountered. The data was then used in order to rank particular routes in terms of how pleasurable they were to cycle on, or how polluted they were. The advantage of this scheme is that bicycles are able to access many areas that motorised vehicles are not, and hence a bike sensor network would provide data of interest to pedestrians too.

Ongoing work at Microsoft Research India sees sensing on vehicles as important, but notes that many vehicles, e.g. rickshaws, are not suitable for the traditional "ruggedised laptop" type of deployment. Instead, mobile phones with Bluetooth sensors are used for sensing traffic conditions and levels of stress (as indicated by the levels of honking) [164]. This, along with BikeNet and a similar bike project, MESSAGE [129], shows a trend away from dedicated computing hardware, and towards the mobile phone as a general purpose device. Whilst the remainder of this dissertation assumes that computing infrastructure will be integrated into vehicles, it is equally applicable to portable devices such as smart phones.

2.2.2 Querying VSNs

In addition to the data collection aspects of VSNs, the question of how to *query* such sensor networks has been addressed. Approaches range from sensors uploading all data to a central repository directly, through delay tolerant networking, to fully distributed data storage. Each requires a different type of network infrastructure.

The VEDAS project [130] constructed a real-time vehicle monitoring system that collected data concerning engine performance and uploaded it in real-time over a cellular link. The focus of the work was therefore on algorithms to summarise the data, to decrease the quantity to be uploaded. Because this was specifically designed as a monitoring system, there were no privacy concerns. Moreover, the data from multiple vehicles was not being aggregated together or queried as a whole.

In contrast, the VITP project [63] allowed users to execute queries concerning the data that the vehicles had collected. Queries had return conditions that specified when they were satisfied: for example, a query for the nearest petrol station required a reply from only one node in the VSN, whereas a query requesting a price comparison would stipulate a minimum number of replies. CarTel's data management system [23] also allowed queries to be issued to the VSN with an SQL-like syntax that included the rate at which sensor readings should be returned.

MobEyes [149, 147] was a completely decentralised scheme, where data was never uploaded to a central authority. Instead, sensor data remained on the vehicles that

had collected it. When a query was issued by a vehicle (expected to be a police unit), the vehicular ad hoc network distributed it, and vehicles across the city that were in possession of relevant data replied. This had the advantage of increased privacy, since data was not present at a single location, but had the disadvantage of the overhead required for the relevant data to be located.

2.3 Communication Systems for Vehicles

The telematics applications that have been described above all necessitate communications technologies. Some require *vehicle-to-infrastructure* (V2I) data transfer, whilst others are *vehicle-to-vehicle* (V2V). This Section overviews the difficulties inherent in achieving wireless communication to vehicles, what part ad hoc networks have to play, and what technologies are available.

2.3.1 General Principles of Radio Communication

Wireless communication for vehicles is complex for three main reasons. Firstly, the environment in which vehicles move (particularly cities) has many (radio) reflective surfaces. Secondly, vehicles travel at a wide range of speeds, resulting in variations in the way communications are disrupted. Finally, radio frequency (RF) interference is common from both in-car sources and other nearby transmitters.

2.3.1.1 Multipath Effects

In an environment where there are no obstacles between two communicating nodes, they are said to have *line of sight* (LOS). Moreover, when there are no objects off which the transmitted signal can reflect, there is only one path that radio waves travel along between the two nodes. If, however, there is a reflective surface nearby, some waves may reach the receiver that have travelled via a reflective path. Such waves will take longer to arrive at the receiver than those on the direct path, as shown in Figure 2.1. Hence, the two waves will interfere with each other. Many reflective surfaces, such as buildings or vehicles, will result in a large number of paths between the communicating nodes. Interference due to such *multipath* effects can result in the receiver incorrectly decoding the transmission.

In order to combat multipath, the time length (period) of each transmitted symbol can be increased. If a symbol is sufficiently long, then waves that have not travelled on the direct path arrive at the receiver whilst the same symbol is still being received from the direct path. The interference is therefore between waves conveying the same symbol, and hence the probability of the receiver decoding the symbol incorrectly is reduced. This lengthening of symbol period is used in Orthogonal Frequency Division Multiplexing (OFDM), where many carrier wave

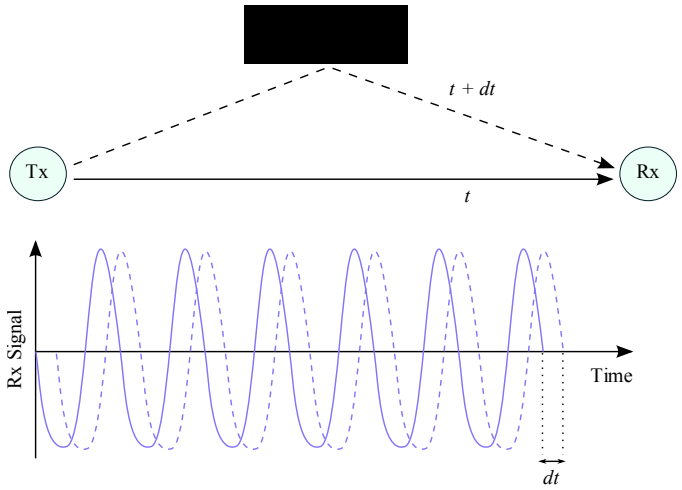


Figure 2.1: When a transmitter (Tx) outputs a signal that is subject to multipath, the receiver (Rx) hears more than one version of the signal, each displaced in time. These can interfere with one another destructively, causing errors in reception.

frequencies are used in parallel to transmit data, each using very long symbol periods. Hence, OFDM transmissions are more resistant to multipath than others that use short symbol periods on a single carrier wave.

Multipath effects are one reason why it is very difficult to predict the coverage of a particular transmitter in a city environment. Buildings can cause rays to diffract as they pass close to them [79, 201], whilst different materials absorb radio energy to varying degrees [69]. Foliage can also significantly affect propagation, with losses of 17 dB being reported for the 5 GHz band when wind caused trees to sway [218]. To accurately simulate propagation, information concerning all of these factors must be known, which in many cases is not practical. Hence, simulation is useful for obtaining a large-scale model, but measurement is currently the only realistic way of obtaining detailed coverage information.

2.3.1.2 Fading

When several non-LOS multipath components interfere, they cause variations in received signal strength (RSS) that follow a *Rayleigh* distribution. This is characterised by occasional deep fades (where the RSS drops momentarily by a large amount, e.g. 30 dB) amidst shallower but longer-lived fades (e.g. 10 to 20 dB). The probability of deep fades occurring is dependent on the RMS value of the RSS, i.e. with lower RSS values the probability of a deep fade is higher. Deep fades can cause momentary losses on the channel, and can also result in incorrect estimations of the RSS, if based on instantaneous measurements. The locations of deep fades

will vary over time if objects in the environment move. This effect is known as *fast fading*. Fast fading is present even if the inter-symbol interference described in the previous Section does not occur. This is because the superposition of multipath components for the same symbol can combine destructively. The resulting signal strength may be too low compared to the background noise to be decoded.

In many situations there is a component that reaches the receiver that is from a direct beam, in addition to multipath components. This changes the RSS distribution to a *Rician* one. The ratio of the power of the principal component compared to that of the multipath components is termed the K value. As K increases, the channel becomes less likely to suffer from deep fades, whilst when K is zero the channel has a high probability of deep fades (and is modelled by a Rayleigh distribution).

In contrast, *slow fading* is a far less random process arising from shadowing by obstacles and attenuation of the signal as it propagates further. Slow fading tends to vary as the transmitter or the receiver move, whereas the fast fading profile for a given location in an urban environment is unlikely to remain fixed over time. Unless simulations of a given environment are carried out, the attenuation due to slow fading is assumed to follow a log-normal distribution, i.e. the distribution of attenuation values in units of dB is Normal. The variance of the distribution depends on the environment, e.g. an urban canyon and a village street would have different distribution parameters.

In order to model a channel correctly, the value of K must be known. For vehicles, the channel is constantly changing, with the motion of the vehicle causing variations in what paths are available between it and the transmitter, and also changing how well any principal component can be received. Hence, fading is both unpredictable and yet has a significant effect on vehicular communications. Further details concerning fading can be found in [94].

2.3.1.3 Interference

City environments are full of wireless transmitters across the frequency range. The 2.4 GHz Industrial, Scientific, and Medical (ISM) band may be used by any device without a license, provided it conforms to certain power limits. Similarly, the 5.2 GHz band is permitted for use for indoor applications. Vehicular communications that take place in these bands are therefore subject to interference that can cause anything from momentary reception errors (e.g. Bluetooth interfering with WiFi [144]) to signal jamming.

Meanwhile, the increasing quantity of electrical equipment in modern vehicles emits low power interference over a wide spectrum. Depending on the location of the communications equipment, this interference can also cause degradation in the performance of the communications channel.

2.3.2 Differentiating Between RSS, RSSI, CQI, and SIR

An indication of how successfully transmissions from a particular base station can be received is given by the received signal power. In absolute terms, this is often measured with respect to one milliWatt. For convenience, *Received Signal Strength* (RSS) values are therefore measured in dBm, the logarithm of the ratio of their power to one milliWatt. A typical RSS value for an indoor 802.11b/g access point measured from the outside is -70 dBm, i.e. 10^{-7} mW or 10^{-10} W.

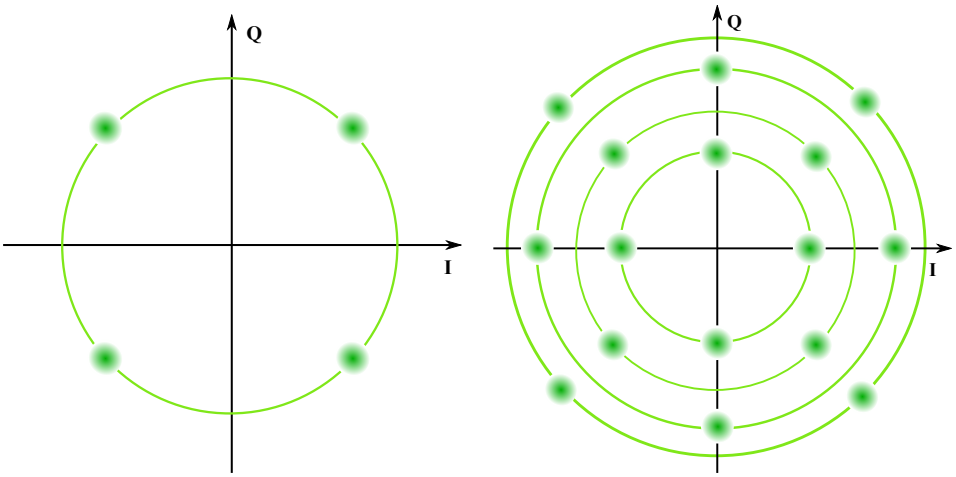
Each receiver circuit has a given minimum received signal power above which it is able to decode the transmission. This is known as the *receiver sensitivity*, and for an 802.11b/g card is approximately -90 dBm. When reporting the quality of a connection, many wireless cards report the *received signal strength indication* (RSSI), which conveys the difference between the real RSS value and the receiver sensitivity. Thus, RSSI is normally hardware specific.

In order to be able to decode a signal, the ratio of its received power versus that of the background noise must be above a certain threshold. This *signal to noise ratio* (SNR) can be calculated from the values in dBm by simple subtraction. Related terms for SNR are Channel Quality Indication (CQI), where the receiver measures how many bits of each correctly decoded symbol were corrupt, and Signal to Interference Ratio (SIR), where interference due to cross-talk from other transmitters is taken into account.

For an in depth description of the different terms in use for RSS and RSSI, the reader is referred to [12], which details the specifics for 802.11b/g hardware.

2.3.3 Relating RSS to Throughput

Wireless technologies that have multiple modes of communication, such as IEEE 802.11 and UMTS, vary the modulation schemes they use depending on the quality of the channel between the transmitter and receiver. Less interference corresponds to a lower probability of confusing one transmitted symbol with another, for a given modulation scheme. For example, Quadrature Phase Shift Keying (QPSK) can transmit four different symbols, each having a different phase, as shown in the Argand (constellation) diagram in Figure 2.2(a). Each symbol can therefore convey 2 bits. In order to increase the bit rate of a transmission, more bits per symbol are needed (assuming that the length of the symbols to be transmitted is held constant), and hence a greater number of symbols is needed. The spacing between symbols in the constellation diagram reflects how similar they are in terms of phase and amplitude: the smaller the Euclidean distance, the more likely it is that the addition of a small amount of noise or other interference will cause a receiver to interpret the signal as a symbol other than that which was transmitted. Hence, higher order modulation schemes such as 16-QAM (Quadrature Amplitude Modulation) as shown in Figure 2.2(b) are less robust to interference.



(a) QPSK (b) 16-QAM

Figure 2.2: Constellation Diagrams showing how higher modulation schemes have lower amplitude and phase spacing between symbols than lower order schemes.[2]

One mechanism that enables transmitters to pick the modulation scheme to be used is the RSS reported by the receiver. For UMTS, the mobile node reports the SNR to the base station 1500 times per second, thus ensuring that any changes in quality are rapidly observed and acted upon. Similarly, 802.11x receivers select the bit rate to be used based on the RSS of the access point they are connecting to, and modify this selection as the RSS changes. Such approaches are logical in that a higher SNR (or, assuming an approximately constant noise power, a higher RSS) will mean that a modulation scheme's symbols can be more closely spaced in the constellation diagram, as interference is less likely. As the separation between the transmitter and receiver increases, the SNR is likely to fall (if nothing else, the signal will suffer increasing attenuation), and hence the modulation scheme selected will change to a more robust one.

In addition to selecting a modulation scheme, a transmitter must also determine what code rate should be used. This is normally expressed as the number of transmitted bits, n, as compared to the number of message bits, m, per block, giving an m/n code rate. The lower the code rate, the less the redundancy of the code, and hence the greater is susceptibility to error. In order that the coding rate may be

[2]The diagram given for 16-QAM shows the circular, rather than more traditional rectangular, form, for illustration of how maximum spacing between symbols is desirable. The rectangular version, where 4 symbols are located in each quadrant of the diagram in the shape of a square, is more common because it is more easily transmitted using two pulse amplitude modulated signals on quadrature carriers, despite not achieving optimal symbol separation.

varied, yet the same decoding hardware used, *puncturing* is used. Here, a known subset of the n output bits are removed before transmission, thus increasing the overall bit rate. Greater degrees of puncturing have an equivalent effect to decreasing the code rate. Hence, for both UMTS and IEEE 802.11, both the modulation scheme and the code rate are selected depending on the SNR.

Because modulation and coding schemes are inherently discrete, there will be a range of RSS or SNR values for which a given scheme is used. Hence, whilst signal strength values at a particular location may vary about a particular mean, provided that such variation is not too great, it is likely that for the majority of the time the same modulation and coding schemes will be used by a transmitter at that location, when communicating with a given base station. Previous work concerning IEEE 802.11 has established such a link [105, 168], and third generation cellular networks using UMTS HSPA or GSM EDGE also show such dependence [58]. Therefore, the remainder of this dissertation assumes that signal strength is a direct indicator of what throughputs can be expected at a particular location.

2.3.4 Vehicular Ad Hoc Networks

Two connection paradigms are proposed for vehicular communications. Using infrastructure-based networks such as the cellular network, where communication is V2I, is one possibility. Another utilises some V2I transmissions, but a majority of V2V communication. The latter involves forming a *vehicular ad hoc network* (VANET), where information is propagated by hopping between the vehicles that make up the network.

VANETs have been the subject of much theoretical research due to a number of challenges inherent in their design [172, 19]:

- **Unpredictable vehicle density:** in some areas the separations between vehicles will be too great for V2V communication to be possible, thus partitioning the VANET.
- **Critical mass of vehicles required:** unless the technology the VANET is based upon reaches a high enough market penetration, too few vehicles will be available to make the network operational. On the other hand, the attractiveness of such a network is rooted in its usefulness. Hence, making such networks useful even when penetration is low is a key challenge.
- **Specialised routing algorithms required:** VANETs are subject to significant churn, as nodes move out of range of one another. The network topology changes rapidly as vehicles move at high speeds. Meanwhile, the best route to a vehicle is dependent on its location, which requires an entity in the network that acts as a location service that is frequently updated by all vehicles.

One project of particular significance is the FleetNet trial [166], where a real deployment of a VANET was carried out. Four cars were used in the testbed. The throughputs achieved were approximately 450 Kbit/s when traversing three mobile hops, as compared to 1800 Kbit/s when only one hop was used. This suggests that ad hoc networks are not well suited to high data rates over many hops.

The Network-on-Wheels project [76] also carried out real vehicle testing, creating a software platform that differentiated between safety and application data forwarding, implemented routing that was location-aware (geocasting), and protected against broadcast storms when large numbers of vehicles were in one place. The successes of the project were in safety-related applications, such as warning drivers to defer right of way to oncoming emergency vehicles at intersections, rather than large data transfers.

Whilst vehicular ad hoc networks do not appear to be suited for high throughput real-time applications over multiple hops, they have successfully been used for carrying data on a *store and forward* basis. Here, nodes carry data potentially long distances, until encountering another node that is likely to carry the data closer to its destination. Thus, network latencies are potentially very high, but such schemes are of great utility where no other network infrastructure exists. DakNet [188] is one such project, where buses (in India) and motorbikes (Cambodia) are used to ferry data from kiosks (such as those provided by KioskNet [204]) in villages to cities. DieselNet [21] is another example of bus ad hoc network, which was set up solely to measure such a system's performance. In particular, the project proposed an algorithm to determine which data should be scheduled for transmission to another bus, and which should be dropped when there was a lack of storage remaining.

Other projects have concentrated on the use of VANETs for more commercial purposes. In particular, Fleanet [148] was proposed to allow drivers to buy and sell goods over a VANET by matching offers and requests over the network. Similarly, AdTorrent [170] was conceived as a mechanism for location-specific advertising to be conveyed to drivers over the ad hoc network. Large files (such as video trailers) were cached in various nodes around the network in order to ease downloading. It remains to be seen how useful such applications are: Fleanet would be competing with existing Internet auction sites, whilst AdTorrent does serve the location-based advertising market, but it is not clear why short adverts could not be served directly from hotspots close to the road. Hence, it is the author's view that there are currently no compelling commercial applications for VANETs, excepting in the areas of safety and connectivity provision in the developing world.

2.3.5 Securing Vehicular Communication

Many ITS applications, particularly those that are safety-related, are required to ensure that communications are secure. Whilst encryption can ensure that data passed between two entities cannot be eavesdropped, other aspects are not so simple.

Privacy: vehicles that broadcast their identifier and location continually are easily tracked without explicit consent. However, such details are necessary in applications such as intersection collision avoidance [64].

Authenticity: it should not be possible for a vehicle to transmit a beacon that includes a spoofed location fix. Similarly, the identifier of the vehicle should be secured to avoid impersonation. Secure positioning services have been proposed for GPS [141] and using verifiable RF multilateration [27].

Integrity: the contents of each transmitted message must not be susceptible to modification by third parties. Man-in-the-middle attacks, such as those where messages are replayed (and hence shifted in time), or where one vehicle masquerades as another, should be difficult to achieve.

In addition, further issues to be considered are jamming (where messages might not reach any recipient) and denial of service attacks (e.g. where the channel is constantly occupied by a malicious transmitter). An in-depth treatment of this area can be found in [191]. This dissertation will not further examine the area of security for vehicular communication, but instead how such communication may be optimised. However, the security mechanisms proposed elsewhere are as applicable to the system developed in this work as to any other concerning vehicular communication, and nothing proposed here precludes their use.

2.4 VANETs Versus Infrastructure Networks

As described above, much research has examined how vehicular ad hoc networks might be used. Safety applications are the most important, with vehicles communicating their positions and speeds to their neighbours for collision avoidance. However, this is essentially one- (or perhaps two-) hop communication, rather than ad hoc networking on a large-scale. The proponents of this large-scale paradigm list several benefits, each of which is outlined below:

- **Speed of propagation of information.** For one or two hop communications, an ad hoc network offers direct transfer of data, and hence low delay. For longer distances, routing complexities and areas of the road where there are no vehicles can mean very slow propagation of information. A simulation for the SOTIS traffic information dissemination system calculated a delay of 27 minutes for a distance of 50 km [243]. This is far slower than utilising cellular links.

- **Cost of deployment.** It is argued that once the network nodes are deployed in vehicles the usage cost of the resulting ad hoc network is zero. However, services such as locating a particular vehicle for routing, and a certification authority to track message provenance, are not cost-free, but required for

many ad hoc network applications. Meanwhile, the cost of data transmission over cellular networks is declining, with some operators now offering fixed price contracts for unlimited data transfers.

- **Infrastructure not required.** In areas where there are sufficient vehicles to form an ad hoc network it can be argued that cellular infrastructure is unnecessary [249]. However, areas of low population will still require connectivity using cellular (or similar) coverage, and hence vehicles will also need a separate network interface for this different technology. This was shown by Huang *et al.* when examining whether an ad hoc network could be used for taxi dispatching [111], and Wu *et al.* concerning message passing down a highway [246]. Linked to this is the observation that cellular providers have already deployed practically ubiquitous coverage for voice calls. Moreover, where VANETs work best (high traffic densities) are where cellular networks are also most dense. Hence, the infrastructure is already deployed in those areas that a VANET might usefully serve.

Thus far, vehicular ad hoc networking deployments have suffered from a lack of take-up from automotive manufacturers and concerns over security. In addition, the throughputs achieved from real VANET deployments such as Fleetnet suggest that throughput decreases markedly as the number of hops increases [166]. Thus, VANETs are unlikely to have the capacity to support bulk data transfers, and hence be used for small messages (such as those for safety applications).

The Infostations model proposed by Frenkiel *et al.* [81] envisages islands of connectivity, i.e. it lies between an infrastructure-less VANET and an infrastructure-heavy cellular network. Today, heterogeneity in network technologies effectively means that such islands exist, with hotspots of high throughput WiFi or HSPA cellular connectivity, (even though the latter is provided by an infrastructure-heavy cellular network). An awareness of where the islands are located becomes enormously important in order that vehicles can, for example, request data to be downloaded to the island ready for transmission to the vehicle. In addition, decreasing the time taken for registration/connection to the network is important: the DriveThru Internet project investigated how such times could be shortened by an in-car proxy automatically authenticating with each network, rather than requiring manual intervention [181]. MIT's CaberNet protocol [72] for uploading sensor data from vehicles similarly includes QuickWiFi, which uses short timeouts in order that disconnections are quickly detected, and a TCP proxy that infers whether packet loss is due to congestion or temporary interference on the wireless link.

In the light of the above, this dissertation concentrates on methods to allow vehicles to better use the multitude of infrastructure-based islands currently deployed (V2I), and does not further examine the rôle of V2V ad hoc networking in bulk data transfer for vehicular communications.

2.5 Available Technologies

Many different technologies have been proposed for V2V and V2I communication [154, 135], some already deployed and others in standardisation. In this Section, an overview is provided of what is available, before focussing on two specific off-the-shelf technologies: UMTS and IEEE 802.11.

- **Satellite:** one-way broadcast communication from satellites is widespread, well-known examples being the GPS [6] (using low earth orbits) and more recently digital radio [51]. Portable Very Small Aperture Terminals (VSAT) allow two-way communication with geostationary satellites, but suffer from high latencies [159]. When a mobile terminal is transmitting, signals are received by the satellite and then sent to a ground station. If communication is to take place between two VSATs, the signal still passes via a ground station before being resent to the satellite and thence to the receiving VSAT, thus incurring even greater latency. Upload throughputs are between 64 and 128 Kbit/s, whilst download throughputs can be up to 438 Kbit/s [75]. Whilst satellite connectivity is ubiquitous[4], its low throughputs and high latencies are unlikely to be useful for two-way vehicular communication. However, it will continue to be effective as a broadcast technology for data distribution.

- **GSM/GPRS:** considered the 2nd generation of cellular network technologies, the Global System for Mobile communications and its associated data bearing General Packet Radio Service run in the 900 and 1800 MHz frequency bands. The system is capable of raw throughputs of between 56 and 114 Kbit/s, depending on the number of channels a mobile node is permitted/capable of using. The highest throughputs are only possible with enhanced GPRS, known as EDGE, which uses adaptive modulation (i.e. the data rate achieved depends on the signal strength at the mobile node). GPRS systems are popular for low throughput telematics applications, such as fleet monitoring, or road pricing (e.g. the German [123] and Swiss [138] heavy goods vehicle charging schemes). In addition, the low frequency used by GSM enables long propagation ranges, which enables coverage to be almost ubiquitous in many nations. High throughput applications are not suited to GPRS, and hence operators are upgrading to 3rd generation technologies such as HSPA over UMTS.

- **UMTS/HSPA:** a major upgrade from GSM is the Universal Mobile Telecommunications System, as specified by the 3rd Generation Partnership Project (3GPP). UMTS has far higher throughputs (multi-Mbit/s) and lower latencies than GSM, and hence is more suited to true mobile Internet access.

[4] At higher latitudes communication with geostationary satellites in urban canyons are problematic, as the satellite may not be directly overhead, but at a much shallower angle. Therefore, whilst theoretically coverage is available globally, users in cities may experience problems.

High Speed Packet Access is a development of the standard which enhances IP throughputs. These technologies are described in detail below.

- **IEEE 802.11:** the 802.11 family of standards (referred to as 802.11x) are intended for local area networking. The physical layer can be infrared, 2.4 GHz, or 5.2 GHz. Various amendments have specified different modulation schemes, allowing raw throughputs of up to 54 Mbit/s for 802.11g. The 802.11a amendment is the basis for intervehicular communication standardisation efforts, with various experimental evaluations having been carried out. A detailed overview is given below.

- **IEEE 802.16 WiMax:** conceived as a fixed broadband wireless access technology, WiMax runs primarily in the 5-6 GHz frequency band (although the standard covers 2-11 GHz). Its main purpose was to provide metropolitan area networking, as opposed to local area provided by 802.11x. It uses ATM-like packet scheduling, where subscribers have assigned transmission slots, thus decreasing time spent contending for slots. Raw throughputs of up to 130 Mbit/s are proposed for nodes close enough to the base station to use 64-QAM modulation. Higher ranges (10s of kilometres) can be achieved at reduced bit rates. Recently, a mobile version of WiMax has been standardised [176], which will be capable of raw throughputs of 15-35 Mbit/s (depending on channel width) [203]. Few experimental results are available to evaluate this claim, with some simulations reporting a throughput of less than 6 Mbit/s [223]. Hence, whilst WiMax may well be a realistic competitor to cellular deployments, this dissertation does not consider it further.

- **IEEE 802.20 MBWA:** in 2002, the IEEE began the standardisation process for Mobile Broadband Wireless Access. The aim is to provide an all-IP wireless interface that operates below 3.5 GHz, at vehicular speeds of up to 250 km/h, and at throughputs of greater than 1 Mbit/s per user. The original purpose of the standard was to replace cellular systems that ran IP over a system designed for voice calls, and did not offer high throughputs. Other systems such as WiMax were not originally intended for high mobility. The need for 802.20 is now less clear, given that cellular systems are increasing in throughput and becoming more data-oriented, whilst the standardisation of a mobile version of WiMax, 802.16e, is currently in progress. Nonetheless, 802.20 also intends to offer antenna diversity, and the capability to work at very high speeds where cellular systems fail (in particular on high speed trains), ubiquitous availability, and high spectral efficiency [134]. The standard [113] was only approved in June 2008, and thus no hardware was available for testing.

- **Bluetooth:** a short-range technology (up to 100 metres) used for networking several personal mobile devices. Bluetooth utilises the 2.4 GHz frequency band, using frequency-hopping spread spectrum transmission. Raw data rates are of the order of 1 Mbit/s. The principal difficulty with Bluetooth is the time taken for it to detect other peers, and then pair with them. In addition, in the author's experience, Bluetooth software stacks under Linux are not stable. Whilst quoted ranges are 100 metres for Class I devices, the author's own experiments found that one device mounted on the front of a vehicle and another on the rear could not communicate reliably. Other work using external antennas has proven more successful, with V2I connectivity established at a speed of 100 km/h for 18 seconds [167]. However, simulations have also shown that TCP connections over Bluetooth perform poorly if two Bluetooth channels are in use simultaneously [211]. Thus, it is not clear whether Bluetooth has the range or scalability for vehicular communication. Moreover, there do not appear to be real advantages over more proven technologies such as IEEE 802.11x.

- **Millimetre wave:** one project that demonstrated millimetre wave technology was MILTRANS [96]. This used frequencies in the 63-64 GHz range, with an IEEE 802.11a MAC layer. Test drives showed a range of up 170 metres (or 140 in fog), with data rates of up to 18 Mbit/s at vehicular speeds of up to 210 km/h. These frequencies have the disadvantage that they are highly dependent on line of sight transmission, and are readily absorbed by water vapour and oxygen. In addition, the technology is not currently available. The rates achieved are similar to 802.11g, and therefore there appear to be no compelling advantages to millimetre wave, other than the availability of the necessary spectrum.

A useful comparison of 802.11, Bluetooth, and WiMax can be found in [142], whilst for further general information on wireless networking technologies [236] is recommended.

2.5.1 Overview of UMTS

The Universal Mobile Telecommunications System is a third generation cellular network technology. It is based on Wideband Code Division Multiple Access (W-CDMA), where multiple nodes may transmit on the same frequency simultaneously, as shown in Figure 2.3. Their signals are distinguishable by the usage of pseudorandom *spreading codes*, which spread each user's data over a wide bandwidth [2], as shown in Figure 2.4. Depending on the rate (bits/second) of the spreading code, different throughputs are supported. This allows a base station to change users' codes based on their throughput demands, and thus use spectrum efficiently. The spread signal is used to modulate a carrier wave (normally in the

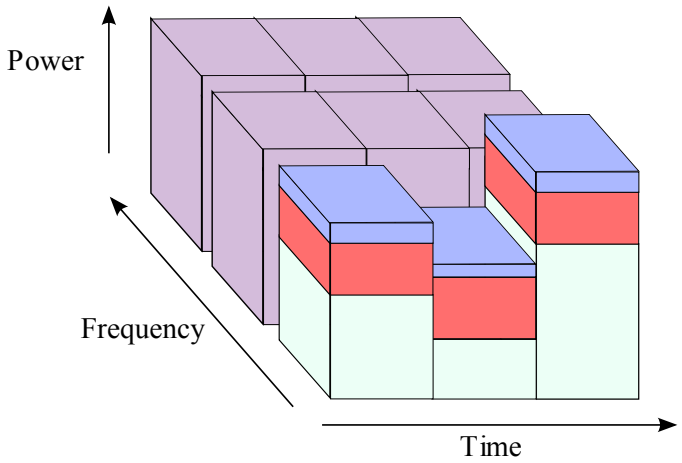
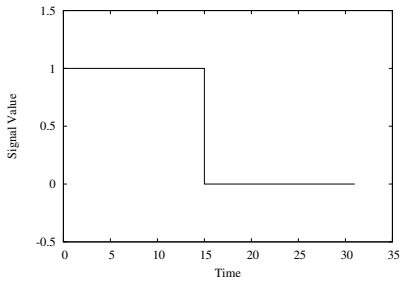


Figure 2.3: How channels are allocated in the time-frequency-code space for W-CDMA. All users from one provider transmit on the same frequency. Each user's power varies over time, as controlled by the base station. Each user has a different spreading code (different colour), giving between 8 and 384 Kbit/s depending on the demands of the traffic [108].

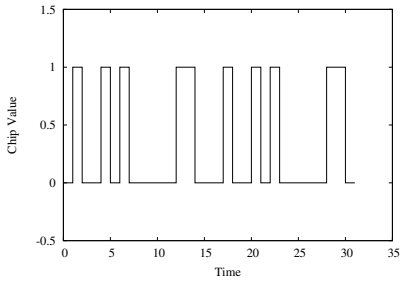
2100 MHz band). The modulation scheme used for UMTS is QPSK, though HSDPA (see below) uses 16-QAM for higher data rates.

UMTS's principal differences from GSM are its wider channels (5 MHz compared to 200 kHz), its use of CDMA instead of GSM's time division on each different frequency allocated to a base station, and its rate of power control (1500 updates per second versus two). A final important difference is that UMTS is entirely packet-, rather than circuit-, switched. This results in far more efficient multiplexing of different users' streams, rather than the wasted capacity inherent in circuit switching with variable rate data streams.

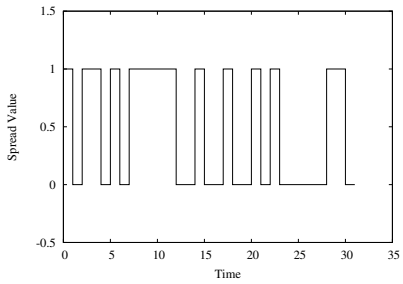
UMTS's rate of power control is significant because of the need to prevent each mobile terminal's signal swamping that of the others at the base station. Because all mobile nodes operate in the same frequency band, if all nodes transmitted at equal powers, a node close to the base station would drown out the signal of a node far away. Thus, the base station issues commands to all mobile terminals to ensure that the received power per bit at the base station is equal for all nodes. The power output by the base station also varies: when there are mobile nodes at the edge of the cell's coverage, the base station increases power slightly in order to compensate for when the mobile node is in a deep fade. At other times, the base station reduces power to decrease inter-base station interference.



(a) Original Signal



(b) Chipping Code



(c) Spread Signal

Figure 2.4: Combining a data signal with a higher bit rate chipping code to produce the spread signal.

The wideband signal produced by UMTS transmitters has another advantage: the multiple propagation paths that the signal travels can be distinguished from each other more easily using a Rake receiver, and hence fading (as described in Section 2.3.1) can be better mitigated against. This leads to increases in performance.

Release 5 of the 3GPP standard for UMTS introduced High Speed Downlink Packet Access (HSDPA) [1]. This uses two important concepts to increase packet throughputs: adaptive modulation and coding, and fast physical layer retransmissions[5]. Adaptive modulation and coding is used in place of variable length spreading codes *and* fast power control. Instead of increasing or reducing transmit power, the base station changes transmissions for users to use more or less robust modulation schemes (thus decreasing or increasing their throughputs, respectively). Users close to the base station have higher signal to noise ratios, and hence use higher throughput (but less robust) modulation schemes than those further away. Meanwhile, Hybrid Automatic Repeat reQuest (HARQ) allows errors to be corrected by multiple copies of a packet being sent using progressively more robust encodings. Once the HSDPA link layer acknowledges the error-free reception of a packet, the next is transmitted, i.e. stop-and-wait ARQ is performed. This is preferable to transmitting streams of packets with the mobile terminal reporting which ones have not been correctly received, as storing all copies of partially received packets can require large memory overheads in the mobile device.

The theoretical maximum raw throughput for HSDPA is 14.4 Mbit/s, but in practice this is normally limited by operators in order to provide for a greater number of users. Typical TCP throughputs on the Vodafone HSDPA network around Cambridge, UK are 1.3 Mbit/s. This compares to raw data transfer rates over UMTS of 384 Kbit/s.

Release 6 of 3GPP defines Enhanced Uplink, otherwise known as High Speed Uplink Packet Access (HSUPA), which provides for a maximum raw uplink throughput of 5.76 Mbit/s. Like HSDPA, HSUPA uses adaptive modulation and coding techniques. Network operators are currently deploying this technology, and hence experimental results are not yet available.

For further details of the 3GPP standards, an excellent overview can be found in [108].

2.5.2 Overview of IEEE 802.11

The IEEE 802.11 standard [114] provides a specification for the creation of wireless local area networks. Three different physical layers were originally defined: infrared, frequency hopping spread spectrum (FHSS), and direct sequence spread spectrum (DSSS). Further amendments to the standard utilised orthogonal frequency division multiplexing (OFDM). Few products using the infrared physical

[5]These concepts are also used to provide the enhanced throughputs provided by GSM EDGE.

layer were produced, whilst the FHSS scheme was rate limited to 2 Mbit/s and hence was quickly superseded by DSSS. Therefore, only DSSS and OFDM will be considered here; further details can be found in [87].

The most well known member of the 802.11 family is 802.11b. This utilises the 2.4 GHz license-free band, and has a maximum raw throughput of 11 Mbit/s. Similar to CDMA (described in Section 2.5.1), the DSSS modulation used in 802.11b combines the data to be sent with a spreading code, generated by Complementary Code Keying [186]. The spreading code can be chosen to result in either four or eight transmitted bits per input symbol, depending on the desired rate and robustness. Once spread, the signal is used to modulate a carrier wave using either Binary Phase Shift Keying (BPSK) or QPSK. Typical TCP throughputs are 6 Mbit/s.

In 2003 the 802.11g amendment [116] was proposed, which offers raw throughputs of up to 54 Mbit/s. The key difference is the usage of OFDM, which provides added resistance to multipath and increased data rates, due to the use of 52 subcarriers of different frequencies in the 2.4 GHz band. Each subcarrier can be modulated using BPSK, QPSK, 16-QAM, or 64-QAM, and its code rate varied using puncturing, depending on the data rate and robustness desired. The throughputs achievable are contingent on whether an AP is to support legacy 802.11b devices ("mixed mode"). If only 802.11g devices are to supported, TCP throughputs can reach 20 Mbit/s.

A less used amendment, but important from a vehicular perspective, is 802.11a [115]. This is very similar to 802.11g, but runs in the 5.2 GHz spectrum, and is thus subject to less interference as fewer consumer devices use this band. It uses OFDM modulation and is capable of raw throughputs of 54 Mbit/s. In the United States, the Dedicated Short Range Communication (DSRC) standard for vehicular communication will utilise 802.11p technology at 5.9 GHz. 802.11p is broadly similar to 802.11a, but specifies control, service and safety of life channels (seven channels of 10 MHz in total), in order to separate different classes of traffic [70]. Meanwhile, the IEEE P1609 Wireless Access for Vehicular Environments (WAVE) standard will specify how layers above the physical and MAC function. WAVE defines the services and messages that may run and be emitted by a vehicle's onboard unit. Whilst the standards are in draft, the performance of the equipment in vehicular environments is closest to that of 802.11a. Hence, part of this dissertation evaluates V2I communication using this technology.

Other significant amendments to the 802.11 standard include 802.11n, which will utilise multiple antennas (MIMO) simultaneously in order to exploit spatial diversity, and higher order modulation schemes (256-QAM) to increase throughputs to greater than 100 Mbit/s. Comprehensive details can be found in Chapter 15 of [86]. As this technology has not yet been standardised, this dissertation will not consider it further.

All amendments to the 802.11 standard use adaptive rate selection, where the transmission rate (and hence the modulation and coding scheme) used is dependent on the channel quality. Thus, in the case of a vehicle passing an access point, the rate

selected will increase up to a maximum (when the vehicle is level with the AP), and subsequently decrease. There is, however, a delay in the selection of a new rate, until the transmitter has confirmed that the signal strength's value is not due to a transient variation (such as entering a short, but deep, fade). Knowledge of the coverage areas of access points can enable rate selection to be carried out more rapidly, thus achieving greater data transfers.

Finally, it should be noted that higher layer protocol throughputs, such as for TCP, are significantly lower than the raw data rates often quoted for the above technologies. The reason for this is that the overhead of encapsulation is significant. For example, a TCP or UDP packet must be placed in an IP datagram, and subsequently in a MAC layer frame for output by the radio transmitter. Hence, maximum UDP data rates for 802.11a are in the region of 30 Mbit/s [31]. For a comprehensive overview of the protocol's MAC layer, see [190].

2.6 Handovers in Heterogeneous Networks

As wireless devices have proliferated, users have come to expect near-ubiquitous connectivity. This goal is coupled with the need to provide connectivity not only for stationary users, but also highly mobile ones; this is particularly true when considering users located within vehicles. As a node moves, the network or base station it is attached to will change, as it transitions between different areas of coverage. Each change is termed a *handover*. In today's world of vast numbers of wireless network deployments (Section 1.2.2) of a wide array of technologies, network heterogeneity is the norm, rather than the exception. This is in turn reflected in the number of wireless devices on sale that have multiple network interfaces, e.g. a mobile phone that uses GSM, UMTS, Bluetooth, 802.11b/g, and contains an FM radio receiver. In addition, network coverage areas are becoming smaller in size to utilise spectrum more efficiently, and achieve higher throughputs. Such conditions mean that a hierarchy of networks exists (see Table 2.2), which Stemm and Katz termed a wireless overlay network structure [215]: ideally, users would seamlessly perform handovers between different networks as they move.

2.6.1 Basic Definitions

Achieving seamless IP-based mobility is encapsulated in the principle of Always Best Connected (ABC) [97], where the "best" network is selected for use whenever there is a choice. In order to allow such mobility, there must exist a mechanism for a mobile node (MN) to update those hosts it is communicating with (*correspondent nodes*, or CN) with the details of what network it has moved to. The simplest scenario is when the mobile node starts a connection in its *home network*, and then moves to a *foreign network*, where it obtains a different IP address, known as the *care-of-address*, CoA.

Technology	Throughput (Mbit/s)	Range (km)	Source
Satellite	0.438 down, 0.128 up (Raw)	$\frac{1}{3}$ of earth's surface	[75, 159]
802.16 WiMax	10 (TCP), 15 (Raw)	10	[253]
UMTS HSPA	1.28 (TCP), 3.0 (Raw)	5 (estimate)	OM, [58, 124]
GSM GPRS	0.04 (TCP), 0.56 (Raw)	<35	[30]
802.11b	6 (TCP), 11 (Raw)	0.25	OM
802.11g	36 (TCP), 54 (Raw)	0.1 (estimate)	OM
UWB	600-1000 (Raw)	0.005	[216]

Table 2.2: Typical throughputs and ranges available from different wireless network technologies, illustrating how these quantities are traded off in the concept of a wireless overlay network. OM stands for "own measurements".

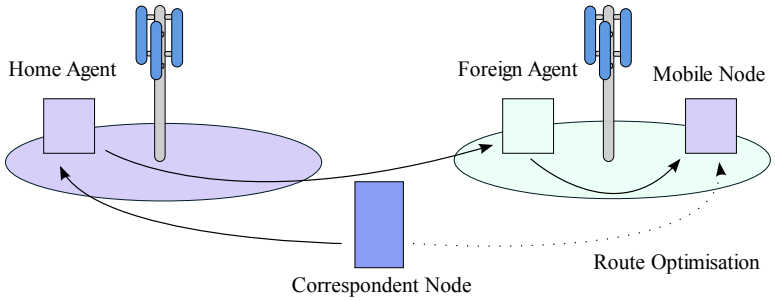


Figure 2.5: Different terms in Mobile IP. When the mobile node is in a foreign network, the HA and FA forward packets on to it from the CN. Route optimisation can be performed for reduced latency.

A variety of mechanisms exist for allowing a CN to continue to communicate with a mobile node when the latter obtains a new CoA. In MoquitoNet [11], the aim was to ensure that only software on the mobile node and in its home network would need to be modified: a *home agent* (HA) received packets destined for the mobile node's home IP address, and tunnelled them to its CoA. In contrast, the IETF Mobile IPv6 specification [119, 67] specifies the use of a *foreign agent* (in addition to the HA), where a dedicated host in the foreign network receives packets destined for the mobile node, and forwards them on. One advantage of this approach is that the FA can forward packets on to another FA in a new foreign network, should the mobile node move. These scenarios are shown in Figure 2.5.

Decreasing the impact of handovers is a well researched area. One problem with using home or foreign agents is the increased latency incurred when packets are forwarded via them to the mobile node. Mobile IPv6 includes a *route optimisation* scheme, whereby the mobile node can choose to communicate its CoA directly to a CN. Hierarchical MIPv6 [214] addresses this issue by means of *mobility anchor points* (MAP) that act as HAs for multiple networks. This then means that an mo-

bile node's HA is now always relatively close-by, thus decreasing the forwarding latency. Meanwhile, the time taken for registering on the new foreign network causes packet loss: Fast Handovers for MIP [136] allows the mobile node to keep its old CoA temporarily when it moves to a new network whilst a new CoA is obtained, thus negating this problem.

2.6.2 Difficulties of Handovers

Network handovers can be divided into two classes:

- **Horizontal handovers** take place when a node changes its point of attachment from a base station of one technology to a base station of the same technology.
- **Vertical handovers** are those where the source and destination base stations are of different technologies.

In a hierarchy of networks both types of handover will take place. Horizontal handovers occur in cellular networks between base stations, whilst vertical handovers take place (for example) when a node leaves the cellular network and connects to a local WiFi hotspot.

During a handover, several stages can be identified. Each is described along with the associated time for a handover from 802.11b to GPRS taken from [234]:

- **Detection Period.** The time taken by the mobile node to discover the available network(s), using link layer signalling or the network layer detection mechanism. 0.80 seconds.
- **Configuration Interval.** The interval from the moment a mobile device receives a beacon from the network, to the time it takes to update the routing table and assign its new CoA. 0.01 seconds.
- **Registration Time.** The time that elapses between the notification of the new CoA to the HA (and possibly CNs), and the reception of the first packet at the new technology's interface using the new CoA. 3.00 seconds
- **Adaptation Time.** Vertical handovers imply significant changes to the physical characteristics of the network connection, which affect protocols such as TCP. A delay occurs whilst the mobile node adapts the connection to the new technology, adjusting the TCP state machine parameters (e.g. congestion window size and timeout lengths) [43]. 2.80 seconds.

It is important to note that not *all* handovers are subject to these disruptions. Horizontal handovers within a single provider's cellular network can occur using soft

handover, where frames are transmitted by two base stations simultaneously, and hence no loss occurs. This allows the detection, configuration, and registration to be performed without impacting the connection. If the source and target base stations have the same throughput and delay characteristics, adaptation time will also be negligible. However, if the throughput available is substantially different, disruption due to adaptation will still occur. Vertical handovers, particularly when these are cross-provider, are the most disruptive.

In order to decrease the disruption of a handover, efforts have been made to reduce the time taken for each of these stages. As described above, much of the effort has concentrated on the configuration and registration intervals. In addition, efforts have been made to modify TCP to lessen adaptation time [185], and improve its performance over wireless networks in general [140, 77]. However, given the sheer number of different wireless networks available, the detection and *selection* of an appropriate network, and *when* to handover to it, is also crucial in ensuring that disruption is kept to a minimum.

2.6.3 Reactive Versus Proactive Handovers

Handover schemes that make network selection decisions only when they detect a change are termed *reactive*. The simplest case is when the mobile node becomes disconnected from its current network, and connectivity must be re-established using another network. However, such an approach can be modified to begin looking for networks when the signal strength of the current network falls below a particular threshold, or when a certain percentage of packets are lost. Devices can also continuously scan for new networks, and if one that has better characteristics is detected, handover to it. If a reactive algorithm uses signal strength as a handover indicator, it is vulnerable to making an incorrect decision based on instantaneous signal strength measurements. If averaging of readings is used, it may mean a significant time passes before the algorithm becomes aware of a step change in signal strength [235]. Balancing these considerations is very difficult. Thus, reactive schemes will *not* result in optimal connectivity, as described in Section 1.3.2.

In contrast, a *proactive* handover scheme uses extrinsic information concerning available networks that allows a more informed choice. Such information might include the load on the network, or its coverage area. This enables such an algorithm to force a handover at the optimal location/time in order to achieve the best service (e.g. hand off to a WiFi hotspot despite cellular coverage still being available). Clients can also prepare for a network handover before the first beacon for such a network is encountered, e.g. by beginning to decrease their TCP adver-

tised window size towards zero.[6] Thus, *better use* of networks can be made with proactive handover algorithms.

2.6.4 Reactive Handover Algorithms

A great many reactive handover algorithms of varying complexity have been proposed. Many input parameters to handover algorithms are possible, such as mobile node speed, a combination of uplink and downlink RSS, or bit error rate. Here, an overview of three main classes of algorithm is given, to illustrate the concept. Further details can be found in [224].

- **Always strongest signal:** the mobile node continuously measures the strengths of the signals received from nearby base stations, and utilises whichever has the strongest RSS. This tactic leads to the *ping pong effect*, where a node frequently oscillates between two base stations as their relative signal strengths fluctuate. Incorporating a degree of hysteresis combats this problem. Examples of this approach are the Mobisteer [171] and MAR [192] projects.
- **Maintain until broken:** the mobile node connects to one base station and continues to use it whilst it is within range, regardless of the availability of other networks. This has the disadvantage that the quality of service offered by the original base station may well be worse than that of another whose coverage overlaps the area the mobile node is in. It is, however, not subject to the ping pong effect.
- **Time-averaged hysteresis:** a solution to the ping pong effect, the mobile node chooses the base station to connect to by averaging signal strength readings over a period of time [88], thus combatting momentary drops in signal strength due to fast fading. The level of hysteresis can be fixed or variable, depending on the environment experienced by the mobile terminal [202].

2.6.5 Policy-based Handovers

Which network is chosen and *when* a handover takes place can be calculated using a policy scheme, that takes as inputs information concerning the different networks, and the user's preferences. Examples of such details are the networks' charging regimes, more dynamic factors such as their loads, and harder to define information such as their coverage areas. Users' preferences might include their financial budget, the minimum throughput required, or the perceived impact of a handover; all of these might also be application-specific. In addition, sensed quantities such

[6] One technique for reducing packet loss on handover is for the client's advertised window to be zero when the handover takes place. This then means that the sender temporarily ceases sending packets until the window is increased post-handover.

as the user's speed might also be used. The quantities are combined into a utility function that allows different network choices to be ranked [161].

Various policy-based decision systems have been proposed. POLIMAND [8] was a generic framework proposed within the IETF that caused handovers to be proactively carried out, rather than waiting until a mobile node exited the coverage of the network currently in use. The choice function given was a generic linear weighted combination of (unspecified) factors. The MosquitoNet project implemented and evaluated a similar system, with good results [11]. A more radical approach was taken by Vidales *et al.*, who allowed arbitrary policies to be specified as finite state machines, compiled in the network, and then downloaded to the mobile device. This gave rise to the PROTON policy framework [232, 231], which was deployed as part of a wider investigation into 4G handovers [234].

2.6.6 Policy-based Approaches Requiring Coverage Maps

Many policy-based systems state the need for knowledge of network coverage areas. This allows network selection to take into account how long the mobile node is likely to obtain service from that network. The cost of handover is then compared to the benefit derived from the new network's connectivity. Examples of such systems are outlined below.

Inoue *et al.* [117] introduced the idea of a mobile node reporting its position to a server over an out-of-band channel, with the server then providing a list of available networks at that location, based on a set of user-supplied preferences. Meanwhile, Zhang *et al.* [255] proposed an algorithm for deciding whether to hand off from a cellular network to a high throughput local wireless network, by means of examining whether the penalty of the handover was compensated for by the benefit of the higher throughput of the destination network.

Similarly, coverage maps can be used for preparing to hand over (network selection having already been performed). Van den Wijngaert and Blondia [228] proposed setting up tunnels from the mobile node to the new FA based on the knowledge of what networks the mobile node would be passing through.

Some authors argue that an awareness of which networks border the one a node is currently connected to, rather than the actual coverage areas, can be used. Here, when a node begins to move, the discovery operation is limited to seeking out known networks, as given by the neighbour graph [207], rather than scanning all frequencies. The draft IEEE 802.21 Media Independent Handover standard also takes this approach [55].

More recent work concerning network access for vehicles has also cited a need for predicting handovers based on an awareness of the RF environment [98], whilst a study of vehicular connectivity in a wireless mesh deployment made use of long-term quality scores for each access point in order to affect which base stations to use [88].

In all these cases there is a need for *some* knowledge of the service/coverage area of each wireless network, in order that a proactive handover algorithm can improve connectivity. However, few schemes have been proposed to *generate* such coverage maps. The next Section examines the work carried out in generating RF maps for wireless positioning systems, and argues how even these maps are not well suited to use by proactive handover algorithms.

2.7 Mapping for Wireless Location Systems

Location systems that utilise RF signals are well known, particularly in the form of the Global Positioning System [6]. In their simplest form, such systems calculate their distances from known transmitters (e.g. satellites), and by multilateration obtain an estimate of their position. In the case of GPS, the distances between satellite and user are large enough to use time of flight to calculate distance. Indoor location systems do not have this luxury. Instead, such schemes have concentrated on using the RSS of base stations to infer how far away they are. If a particular model of wireless propagation is assumed (typically: line of sight transmission, and power falling with the square of distance), then readings for three or more base stations will position a user at the intersection point of three circles (each centred on a transmitter). Such an approach was used in the SpotON RFID location system [107]. Crucially, this method requires knowledge of the positions of the base stations.

An alternative approach consists of constructing a database of RSS readings, known as *fingerprints* at a number of locations throughout the area under consideration. Each fingerprint is a vector of the RSS for all transmitters that can be received at that location. When a mobile device subsequently wishes to obtain its position, it can compare the fingerprint it sees with the database, and interpolate those fingerprints that are the best matches in order to derive its location. One early application of this principle was the RADAR location system [9] at Microsoft Research and the Nibble [29] room-level location system from UCLA.[7] This was followed by the Place Lab project [143], which used beacons from multiple wireless technologies for indoor and outdoor location. Such approaches require that the fingerprint derived at each location remains fixed, and that the fingerprint database is sufficiently dense that at least one close match can always be obtained.

Collecting large numbers of fingerprints is labour-intensive. Therefore, a variety of techniques have been proposed in order to generate synthetic sample points for addition to the fingerprint database. The accuracy of these techniques is in part dependent on how complex the RF propagation environment is: in a large open space the generation of such *fictitious training points* may work well. A scenario where multipath is common and changes to reflective surfaces (e.g. doors opening

[7] Though in fact the RADAR project also examined using their own indoor propagation model to generate RSS fingerprints, in order to avoid collecting them.

and closing) are frequent, such methods may not be as appropriate. It is interesting to note that most authors justify fingerprinting (as opposed to using propagation models to entirely generate sample points) because models of propagation are not representative of the environments they are working with. However, the same authors proceed to generate fictitious points by means of interpolation, i.e. relying on the simplicity of radio propagation. This may be justified if the interpolation is over a small range, where it is assumed that e.g. signal strength falls with the square of distance. However, the author has seen no work examining at what point such an argument breaks down. Meanwhile, the Horus WiFi location system [254] explicitly includes mechanisms to combat the small scale variation (fast fading) that the authors claim is inherent in wireless LAN deployments.

A variety of methods exist for attempting to infer what coverage is available at locations other than those explicitly sampled. These are outlined in the Sections that follow.

2.7.1 Weighted k-Nearest Neighbours

With a large enough sample set, a location estimate can be constructed simply by interpolation between known values of RSS at particular locations. The process involves the vector of RSS values measured by the mobile terminal being compared to the fingerprints in the sample set, and those samples that are "closest" (in terms of RSS values) being used in the calculation. The value of k limits the number of samples used, since points far away will in fact make the result *less* accurate. Such an approach requires a very detailed survey to be performed, unless the radio environment is quite uniform.

Hossain *et al.* [110] adopted this method, whilst Li *et al.* [150] also evaluated kriging (another interpolation technique, described in Section 4.4.3) for generating further points. Hence, both collected relatively few RSS points in their initial site survey.

2.7.2 Neural Networks

Neural networks have long been used in pattern recognition applications, and hence seem a good fit for recognising location from signal strength readings. The basic structure of a feed-forward neural network involves three layers: input, hidden, and output. Neurons are activated by a certain level of signal, i.e. their input must exceed a certain value. If it does not, the output of that neuron is zero. Each input arrives at an input neuron, which is connected to all neurons in the hidden stage. The input neurons output copies of their inputs to all neurons in the hidden layer, but first apply a weighting function. This means that the values received at each of the hidden layer's neurons from one input neuron are likely to be different. The hidden layer applies its own set of weights (also known as free parameters), and

feeds into the output layer. The activated state(s) in the output layer are the result. By changing the weights used, the neural network can be made to output the position of a mobile terminal when supplied with the RSS values of APs in range [13]. The accuracy of prediction will in great part depend on the amount of training the network receives. Interestingly, if a network is *over-trained* then its ability to correctly predict for inputs not in its training set decreases. Therefore, when using such an algorithm the number of training iterations needs to be carefully controlled.

2.7.3 Support Vector Machines

Support Vector Machines function by constructing a hyperplane that achieves maximum separation between two sets of points, namely those that satisfy a particular classification, and those that do not. An example is the hyperplane separating the set of vectors of signal strength readings that could be obtained when located in a given room, from the set of vectors that could *not* be obtained in that room. The dimensions of the hyperspace can be further augmented to provide an indicator hyperplane that, for a given vector of signal strength values, predicts which locations could have provided such a set of readings. Such a method is presented by Battiti *et al.* [14], concluding that the positioning achieved is superior to the methods described above.

2.7.4 Unsuitability for Proactive Handover Algorithms

The approaches described above appear to show some success in the field of wireless positioning systems. The question that must be answered is whether they can be of equal utility as regards proactive handover algorithms.

The majority of the methods for wireless positioning rely on a database of RF fingerprints. Such databases are likely to be very large when considering, say, a city, if accurate positioning is to be obtained. The Place Lab project did not regard this as a problem, given the availability of cheap flash memory for portable devices [106]. However, it is not clear that querying such a large database on a resource-constrained device could be made efficient. Moreover, such databases of raw sightings are not in a form that is can be easily queried by a proactive algorithm: what is required is a knowledge of the service boundaries and black spots of networks, rather than randomly positioned signal strength readings. Of course, such data sets could be *processed* to provide such information, but the approaches described above do not perform such processing.

Hence, wireless positioning systems can provide a good source of raw data with which to produce coverage maps, but do not provide maps that are useful for handover algorithms. Chapter 4 of this dissertation addresses this problem.

2.8 Chapter Summary

This Chapter has provided a general background of how Intelligent Transportation Systems have contributed to the possibility of Vehicular Sensor Networks. Moreover, in order to achieve their full potential, such sensor-equipped vehicles require seamless communication services. This Chapter has overviewed the different technologies that are available for vehicular communications, in particular UMTS and IEEE 802.11x, and argued that to use the multitude of available networks reactive handover algorithms are not powerful enough. Instead, coverage maps can enable proactive algorithms to handover when it is optimal to do so. However, the coverage maps produced for wireless positioning systems are not suitable for this task. Hence, algorithms for constructing suitable maps are required.

3

The Variability of Wireless Coverage

IN ORDER to make the vision of sentient transportation a reality, we must first examine the problems inherent in deploying computer and communications technology in the vehicular environment. Whilst simulation of various aspects of this domain is possible, such simulations are unavoidably simplifications of real life, and generally do not consider the interplay of a large number of factors. A case in point is how transport engineering researchers have a tendency to use realistic models of traffic flow, but model wireless communications as having circular coverage areas. Meanwhile, networking researchers attempt to use more realistic radio models, but employ woefully inadequate models of vehicular movement, such as random walk. Given these issues, the decision was taken to deploy a vehicular testbed on which a variety of experiments could be carried out, over a long period of time. This Chapter describes the architecture of the platform, and the results of applying it to two communications technologies, namely IEEE 802.11 and UMTS. Firstly, the degree of consistency that the performance of these technologies has with time and meteorological factors is examined. The remainder of the Chapter focuses on how 802.11 is affected by beacon rate and AP position, and evaluating the performance of 802.11a for the case of a moving vehicle.

3.1 A Platform for Investigating Sentient Transportation

In carrying out research into sentient transportation, The goal was *not* to deploy computing infrastructure unobtrusively in a private vehicle, but instead to provide a *platform* on which research into ITS could be carried out. The motivation for this was to achieve a system that satisfied the following objectives:

- Flexibility to provide support for future addition of equipment
- Robustness enabling a long (multiple years) operational lifetime
- Ease of maintenance and development
- Ability to store large quantities of data
- Ability to autonomously carry out data collection and upload.

These criteria meant that the deployment differed from the many other "intelligent vehicle" projects, notably MIT's CarTel [112], Leeds' Instrumented Car [219] and the Vehicle Performance and Emissions Monitoring System (VPEMS) [174]. These projects have all tended to focus on one particular aspect, such as communications performance, rather than the collection of large amounts of data concerning the environment of the vehicle in[1]. Such data may include communications-related items, but in wishing to investigate how sentient computing paradigms could be applied in this context, it was clear that a far greater variety of sensors would be needed.

Motor vehicles are of course not the only ones that can be used to collect data about the urban environment. In particular, bicyles offer more load-bearing capabilities than pedestrians, whilst being a mode of transport that can cover long distances. The BikeNet [71] and Mobile Environmental Sensing System Across a Grid Environment (MESSAGE) [129] projects are good examples of how bicycles can be used, with the sensing of pollutant levels being a popular application. Accordingly, the majority of the work described in this dissertation is applicable to vehicles in general, including bicycles. The choice of a motor vehicle reflects the desire for a general purpose vehicular computing platform, as well as for providing an incentive for researchers to use the vehicle (thus collecting data), which would not have been the case with a departmental bicycle.

The Sentient Vehicles project (inspired in part by earlier work within the same research group [233]) was conceived as an evolutionary platform, designed to collect a large corpus of data for informing further research into vehicle-related computing, whilst on the other hand allowing this sensor data to be used by sentient computing applications running onboard the vehicle. A small van was purchased, in order that it would have sufficient space for easy system deployment, and yet be usable in the same environments as a normal private car. The salient features of the platform are described in the following sections.

3.1.1 Architecture

3.1.1.1 Computing Infrastructure

The Sentient Van uses a low power computer as a compromise between electrical power consumption and performance. The machine contains a 1 GHz processor mounted on a Mini-ITX form-factor motherboard, with a power consumption of less than 20 W. This is sited in a small rack at the rear of the vehicle, which also contains an auxiliary battery and associated power control hardware, as shown in Figure 3.1. By avoiding the use of the vehicle's own battery, the vehicle is still usable even when the computer is accidentally left switched on. This approach was

[1] Although the Leeds project is an exception, there do not appear to be any publications concerning its results.

in recognition of the fact that it was expected (which transpired to be correct), that ongoing development on the platform would result in occasional mistakes being made.

In addition to the sensor infrastructure described below, an interface was constructed between the vehicle's audio system and the computer, and signals from the control stalk next to the steering wheel were intercepted in order to use this as an input device. For output, two 8" touchscreens were mounted in the front part of the vehicle, whilst a 17" LCD monitor on a retractable assembly was located in the rear, for when development was to take place *in situ*.

3.1.1.2 Network Buses

Modern vehicles utilise communication buses, rather than per-component dedicated wiring, to connect sensors, processors and actuators. The most prevalent technology currently in use is the Controller Area Network (CAN) bus [118], which incorporates various features that make it ideal for safety critical applications. Notably, it guarantees message delivery, and allows messages to be prioritised to ensure that critical messages arrive at their destinations. This is the case even when a high priority message's transmission begins simultaneously with one of lower priority, a scheme known as Carrier Sense Multiple Access/Bitwise Arbitration. For distances of less then 40 metres, the bus has a throughput of approximately 1 Mbit/s.

Whilst the vehicle's CAN bus could theoretically have been used to connect the platform's sensors to the onboard computer, this was not done for two reasons. Firstly, such an approach could (potentially) compromise the safety of the vehicle, which could have been the case were there to have been errors in the platform's software. Secondly, accessing or extending the vehicle's CAN bus to where the sensors were positioned would have been challenging. Therefore, a dedicated CAN bus was deployed throughout the vehicle.

In addition, wired ethernet and a USB tree were deployed. These networks allowed us to provide for higher data rate sensors, such as video cameras, and also provided for devices with USB interfaces to be easily added (rather than requiring a custom CAN interface board to be made).

3.1.2 Onboard Sensors

At the time of writing, the platform includes the following sensors:

- 2 GPS receivers
- OBD-II interface (for obtaining data from the vehicle's CAN bus)

- RFID reader (for reading identity cards)
- Digital video cameras (forward and rear)
- Carbon dioxide sensor
- Temperature sensor
- Humidity sensor
- Barometric pressure sensor
- Inclination sensor
- 2-axis accelerometer
- 2-axis magnetometer (for sensing heading)

Each sensor is sampled at a frequency appropriate to its function: e.g. GPS positions are obtained every two seconds, whilst images are requested from the cameras at a variable frequency depending on the speed at which the vehicle is travelling. All data is logged to text files on the computer's hard disk, rather than using a database system, for three reasons. Firstly, the complexity of most database systems leaves them open to database corruption if the computer is turned off unexpectedly, though transaction support would mitigate against this. Secondly, by using text files it was simple for different applications to obtain sensor data by reading the appropriate file, in contrast to opening a database connection and issuing a query for the most recent values.[2] Finally, text files are more amenable than databases to simple file synchronisation utilities such as `rsync`, which was used to upload sensor data from the vehicle to our laboratory.

After the initial deployment, the flexibility of the platform was illustrated by deploying a subset of it in a rented vehicle, in approximately three hours. This vehicle was then driven from Cambridge, UK to Graz, Austria, and back. The system successfully collected data for the entirety of the journey, thus demonstrating its robustness.

3.1.3 Communications Infrastructure

Another important aspect of the vehicle is the communications infrastructure deployed in it. In addition to the internal network buses described above, network interface cards that enabled communications with wireless networks that the vehicle encountered were also deployed. Initially, an Orinoco IEEE 802.11b card was used

[2] Whilst a more heavyweight middleware solution, involving sensors publishing their values to applications that subscribe to such event streams was considered, simplicity was preferred in order to ensure robustness.

Figure 3.1: The equipment rack at the rear of the Sentient Van.

Figure 3.2: Antennas deployed on the Sentient Van. The white antenna is for IEEE 802.11a at 5.2 GHz, the tall metallic one for IEEE 802.11b/g at 2.4 GHz, and the remaining small black patch antenna is used for GPS reception.

for WiFi, and an Option Globetrotter UMTS card for cellular connectivity. Both cards make use of external omnidirectional antennas (shown in Figure 3.2) with a gain of 7.8 dBi for 802.11b/g, and the standard external antenna (of unknown gain) supplied by the manufacturer for the UMTS card. The latter was subsequently upgraded to a card capable of High Speed Packet Access (HSPA), providing significantly higher TCP download throughputs of up to 1.3 Mbit/s (as compared to 384 Kbit/s for the standard UMTS card). When UMTS service is not available, the cards fall back to using the GPRS data service over the GSM network.

3.1.3.1 IEEE 802.11b/g

As part of the logging process, software was written to query the received signal strength reported by the network cards. For IEEE 802.11b/g, the majority of networks are configured to broadcast beacons advertising their presence approx-

imately every 100 ms. Because there are 11 (overlapping) channels that may be used in the UK, the card changes channel frequently when in scanning mode, to ensure that networks on all channels are detected. On each channel, it broadcasts a probe request frame in order to cause all access points in range to respond with a probe response[3]. However, it is possible for a network to go undetected in the approximately 180 ms that the card spends on each channel, if a beacon is lost or the channel congested. To mitigate this, the logging software uses a daemon that requests the card emit a probe frame (or return the scan results if a probe is currently in progress) each second. The MAC addresses of the access points from which reply frames are received are logged, together with their network names (ESSIDs), channels, encryption level, and the powers of the received signal and background RF noise.

3.1.3.2 UMTS

For logging signal strength readings for cellular networks, a first attempt was to obtain the complete list of base stations (and signal strengths) that were currently visible to the interface card. This would have included details on all network providers. Unfortunately, this type of request causes the card to conduct a network scan that lasts of the order of 30 seconds, which was deemed too long for when the vehicle was on the move. Instead, the card is polled for the signal strength of the base station it is currently registered with. This approach has the downside that the RSS value recorded at any point will (to an extent) be a function of the route taken in order to reach that location. Such a situation arises because the network incorporates hysteresis in order that clients do not "ping pong" between two similarly strong base stations. This means that a mobile terminal can be connected to a base station that is not the one with the strongest RSS value at that location, until the network infrastructure makes the decision that a handover should occur. However, this is offset by the provider used (Vodafone) having a relatively sparse deployment of base stations around the city of Cambridge, combined with the fact that whilst recording RSS values the card in the vehicle never has a call in progress. The latter observation implies that the penalty in changing base-station would have been minimal, as no spectrum resource would be required at the new base station. Hence, whilst there is no hard evidence for this, the results indicate that the values reported are those of the base station with the strongest signal. Moreover, by electing *not* to record the cell base station ID, data is collected concerning what level of service the network decides the client should have. This is logical, since the *network* chooses when the client should perform a handover.

[3] If the card is connected to a network, scanning is far more restricted, as the time the card spends absent from the channel must be small in order for it not to miss frames destined for that card.

3.1.3.3 IEEE 802.11a

For investigating the performance of IEEE 802.11a, an NEC Warpstar 802.11a/b/g card was added to the system, combined with an external omnidirectional antenna providing a gain of 7 dBi for the 5 GHz band. This card was not used for the logging of 802.11a signal strength values, as drivers for doing so were not available.[4] This was not a particular loss, as there are very few networks deployed that utilise 802.11a, at the time of testing. The main reason for using this equipment was that the 802.11p standard for vehicular communication is based on 802.11a. Experiments with this technology are described in Section 3.5.

3.1.3.4 Bluetooth

For Bluetooth, a data transfer was attempted between two Belkin Class I USB dongles, one placed on the bonnet of the vehicle, the other on the outside of the rear door. Despite the fact that this product claims transmission ranges of up to 100 metres, it was found that the set-up could not reliably sustain a connection. Data transfers frequently failed, resulting in the devices needing to be re-paired with each other. Investigations were therefore not pursued with this technology.

3.1.3.5 IEEE 802.16 WiMax

For 802.16 WiMax, brief tests were carried out concerning the feasibility of using such technology for connecting to vehicles (including whilst moving at low speeds), by using an Alvarion BreezeMax base station and subscriber unit. Provided that the vehicle's antenna was located at a height similar to the surrounding trees, it was found that a connection could be achieved over distances of over 500 metres. This was not further investigated, due to spectrum licensing requirements, and hence further results concerning range (which can be expected to be of the order of kilometres) or other parameters are not given here.

3.1.3.6 Usage of Communications

In addition to the logging of RSS values for IEEE 802.11b/g and UMTS/GPRS, the communications infrastructure onboard the vehicle was used for a variety of other purposes:

- **Sensor data upload:** by using the information about which wireless networks were available, it was possible to infer when the van has returned to

[4] Using wireless interface cards in *monitor* mode is a fraught process. Suitable drivers are only available for a few cards, which, combined with the capability of a PCMCIA card to accept a connection from an external antenna, is the reason why Orinoco cards are so popular for this type of application.

base. Having waited a small period of time to ensure that the vehicle was returning (rather than driving away, or passing by), the software then configured the network interface, and synchronised the sensor data with the backup copy on the laboratory file server. This contributed to the aim of making the vehicle's computing infrastructure run without manual intervention.

- **Vehicle tracking:** for security reasons an application was written that uploaded the current GPS position of the vehicle to the laboratory servers using the UMTS link. This then allowed the vehicle to be tracked when it was being driven, and thus might have been of use were the vehicle to have been stolen.
- **Wireless hotspot:** using standard Linux utilities, the vehicle was converted into an Internet gateway. An IEEE 802.11b/g wireless access point was placed in the vehicle, and connectivity provided to clients in and around the van, with traffic passing over the UMTS interface.

Clearly, such applications only scratch the surface of what is possible, given the richness of the sensor data being collected, and the throughputs available from the communications infrastructure deployed.

3.1.4 Obtaining a Large Corpus of Data

In constructing the platform the goal was made to ensure that users of the vehicle would not need to interact with the onboard computing infrastructure, i.e. that data would be logged without any user intervention. In addition, the vehicle was offered for use by any member of our laboratory. These two factors meant that over three years the vehicle has been driven on a wide variety of roads: motorway, rural, and urban. Journeys have been both short and long, with many different drivers, at a wide range of times of day, and in all seasons. The corpus of data built up reflects the large number of journeys made, with 56.4 million readings collected.

3.2 Environmental Effects on Wireless Performance

Using the Sentient Van as a platform, performance data was collected concerning IEEE 802.11b/g and UMTS networks. This consisted of both the RSS measurements logged on all journeys made, and also throughput measurements made in dedicated experiments (which, for IEEE 802.11 concerned the 802.11a amendment, as there already exists a large body of work on 802.11b/g). In order to assess the performance of wireless technologies when used on moving vehicles, it was necessary to first establish whether these technologies had predictable performances for static nodes. Were this not to be the case, attempting to make meaningful conclusions concerning the effects of mobility would not be possible. Therefore, this Section addresses the questions of whether RSS is constant at a given

geographical location, whether meteorological factors affect RSS, and what the relationship between RSS and throughput is.

3.2.1 Related Work

Several projects have collected large quantities of RSS or throughput data for a variety of reasons. The CarTel project carried out a large-scale survey of the performance of the wireless access points (APs) found in the city of Boston, USA [24]. The aim of the investigation was to ascertain what connection quality such APs could provide, but the study did not assess how consistent this was over time. Similarly, the Place Lab project [33] has collected data concerning 35,000 APs in the Seattle, USA area, for use in wireless positioning. Whilst depending on these readings remaining constant over time in order to calculate users' location, no *long term* study was performed to confirm that this was the case. Of course, the evaluation of positioning accuracy carried out for the project does show that this assumption must hold to some degree, but no statistics are provided concerning the magnitudes or causes of variation. Meanwhile, measurements carried out by Microsoft Research at their Redmond campus investigated how well beacons from APs were received by shuttle buses, and detailed how the locations of areas of good or bad coverage ("grey areas") were fixed [155]. However, the data were not sufficient to analyse long term variations. In contrast, the data presented in this dissertation span multiple months and all hours of the day, enabling a statistically meaningful analysis to be undertaken.

Other work whose main goal was not analysis of RSS variation has none the less provided useful data concerning it. In particular, Grade *et al.* note that in their tests of 802.11b, RSS was constant irrespective of speed on all their drives past an AP [92], whilst Hossain *et al.* measured the standard deviation in RSS at a single point in their indoor location testbed to be 8 dB [110]. Interestingly, other work [126] specifically concerning indoor RSS variation found that its distribution was not Gaussian (the resulting distribution was left-skewed). However, when modelled by a Gaussian distribution, positioning algorithm accuracy was still acceptable. This suggests that a Gaussian model is a reasonable approximation (even indoors).

As regards RSS variation due to differences in the hardware used, Hossain *et al*. also measured the RSS at 20 different locations in their testbed [110]. Their graphs show that the RSS trends shown by two measurement devices are similar, and have a constant difference. Meanwhile, Haeberlen *et al.* showed that there is a linear relationship between the RSS values reported by different hardware configurations [99]. Therefore, differences in hardware configuration can be easily compensated for.

In terms of how wireless coverage varies with meteorological effects, it is important to note that such effects are very dependent on the frequency being considered.

Experiments performed at 3.5 GHz found that the most important effect on long-distance links was the quantity of foliage, which was a seasonal effect [46], rather than the actual weather. The relationship between attenuation, A (in dB/km) and rain rate, R (mm/h) is modelled as $A = aR^b$, where a and b are frequency and temperature-dependent coefficients [175]. At frequencies of above 10 GHz, this effect begins to be significant: with a rain rate of 50 mm/h a 1.08 km link suffered 5 dB of attenuation at 23.6 GHz, and 15 dB at 38.9 GHz [101]. Such links have also been used to *infer* rainfall from attenuation rates [163, 162], due to their sensitivity to precipitation. In contrast, at 2.5 GHz, $20^{\circ}\mathrm{C}$, and a rain rate of 50 mm/h, only 0.002 dB/km of attenuation is predicted by the $A = aR^b$ relation [175].[5] Thus, rainfall at the frequencies used by IEEE 802.11 and UMTS is unlikely to have any discernible effect. The only work concerning the effects of meteorology on frequencies of 5 GHz and below reported that heavy rain caused a 4 dB loss over a 28 km link at 2.4 GHz, which is negligible. Other factors did not have any effect, except snow, when it stuck to the antennas [177].

3.2.2 UMTS Variability

In order to assess the variability of UMTS RSS, logging was carried out at a single location over a 2.5 month period between the end of October 2007 and the beginning of January 2008. A test machine was located on the window sill of an office, located approximately 960 metres from the nearest Vodafone UMTS base station. The equipment continuously recorded the RSS as reported by an Option 3G PCMCIA card once every three seconds. Meteorological data for each half-hourly interval was obtained from the University of Cambridge Computer Laboratory weather station[6]. For a few brief periods during the logging either the weather data was unavailable, or the test machine failed, but this did not significantly affect the number of readings recorded, which totalled 1.34 million.

3.2.2.1 Heat Map Representation

Each RSS reading was associated with a meteorological reading by using linear interpolation. This made the assumption that meteorological parameters varied linearly over each half hour interval. Histograms of the data were then plotted for each meteorological parameter, in the form of heat maps. Each cell on the heat map corresponds to a small range of values in RSS and another small range of the relevant weather parameter, e.g. for pressure there is one cell concerned with values of between 1000 and 1005 mBar and values of RSS between -105 and -103 dBm, whilst there is another for the same values of pressure but an RSS reading

[5] This is in contrast to the widely held myth that OH bond in water absorbs microwave radiation best at 2.4 GHz. Instead, the heating effect is due to water molecules rotating as the electromagnetic field oscillates, which will take place at any frequency.

[6] `http://www.cl.cam.ac.uk/research/dtg/weather/`

of between -103 and -101 dBm. The colour of each cell represents the number of readings that fell into that cell's range of values, with lighter colours indicating a greater number of readings. The end result is a histogram with two independent variables, where each cell is a histogram bucket, and the colour the height of the histogram bar.

In each pair of heat maps, the first shows the raw number of readings that fell into each bucket, i.e. it is a histogram of the source data. The second heat map in each pair shows these values normalised according to the number of readings seen within each bucket. This is done by counting the number of readings obtained within each range of the meteorological parameter, dividing the total number of readings collected by this figure, and multiplying the result by the values in all cells that correspond to that range of meteorological values. Such a normalisation is necessary to ensure that correlations of RSS with meteorology are not hidden solely due to the majority of the time being spent experiencing "prevailing" weather. As an example, were RSS to be correlated with temperature, we would expect predominantly low values of RSS at -10 ^{o}C. However, with very few readings at such low temperatures, such trends would not be evident on the heat map of raw data, whereas they should be so on the normalised data heat map. A disadvantage of using normalisation is that it skews the results when a particular bucket has only a few values in it, i.e. trends will appear to be present that are not. Hence, both types of graph are presented for clarity.

3.2.2.2 Meteorological Effects

Figures 3.3 and 3.4 show the interdependence of pressure, temperature, absolute humidity (i.e. the quantity of water in each cubic metre of air), and wind speed on UMTS RSS. Pressure, (Figures 3.3(a) and 3.3(b)) appears to have no effect, with both raw and normalised heat maps showing no correlation with UMTS. In contrast, RSS appears to be negatively correlated with temperature (Figures 3.3(c) and 3.3(d)), the normalised heat map showing this more clearly. Calculating the Pearson product-moment correlation coefficient on the raw data for temperature gives a value of -0.27, indicating a very weak negative correlation.[7] A similarly weak correlation coefficient of -0.23 is exhibited by the RSS versus absolute humidity data (Figures 3.3(e) and 3.3(f)). Finally, wind speed (Figures 3.4(a) and 3.4(b)) does not appear to have any effect on UMTS RSS (correlation coefficient -0.09), though Figure 3.4(a) shows that the majority of the time period under investigation the wind speed remained low, and hence were an effect to be present

[7]The calculation of the Pearson coefficient assumes that both variables are normally distributed. Figure 3.6(a) shows the distribution of the RSS values recorded approximates to normal. For temperature, Figure 3.6(b) also shows an approximately normal distribution.

it may not be evident in these graphs. Overall, given these results, it can be concluded that meteorological factors do not significantly affect UMTS.[8]

3.2.2.3 Temporal Effects

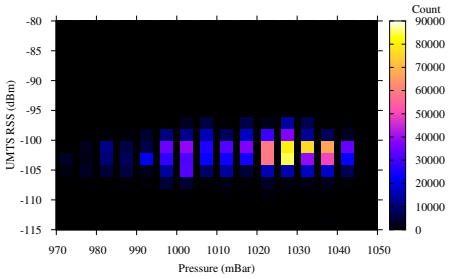
Figure 3.5 shows the effect of time of day on UMTS RSS. Figures 3.5(a) and 3.5(b) show that during working hours (defined as 09:00 to 19:00) the RSS is not as consistently high as during non-working hours. This is depicted by darker colours in the heat map during these hours. The effect is further demonstrated by Figures 3.5(c) and 3.5(d), where the distribution for non-working hours can be seen to be slightly less wide, and somewhat shifted to higher RSS values. The means of these RSS distributions are 5.40 and 5.72 respectively.

The implication of these measurements is that when a base station is under load during working hours, its power output will decrease somewhat due to an effect termed *cell breathing*. This technique is used to decrease the coverage area of a cell in order that subscribers furthest away will connect to a neighbouring base station. In these experiments, the measuring equipment was relatively close (less than 1 km away to a base station). At higher distances the effect may be more evident.

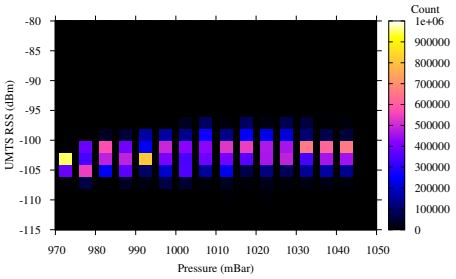
3.2.2.4 Applicability to a City

Separately, data collected concerning UMTS RSS recorded by the Sentient Van when in its home location also shows a Gaussian distribution. The data have a standard deviation of 3.5 dBm, suggesting that 90% of the values will be within 7 dBm of the mean. Data from the Sentient Vehicles project also indicates that UMTS RSS is fixed over time over a wider geographical area. Figure 3.7 shows a heat map of RSS values logged over three years around the city of Cambridge, UK. Darker values correspond to higher values of RSS, and cellular base stations are also marked. Qualitatively, the colour at any one location on the map appears quite uniform, and gradients in RSS can be seen as distance from base stations increases. This reinforces the conclusion that UMTS RSS can be predicted from historical data.

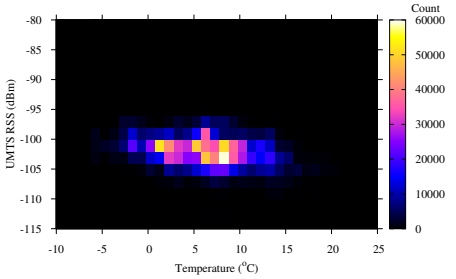
[8] There is the possibility that meteorology might have secondary effects that in turn impact UMTS RSS, which were not measured. Given that the correlations observed were very weak, it can be concluded that such effects are not significant.



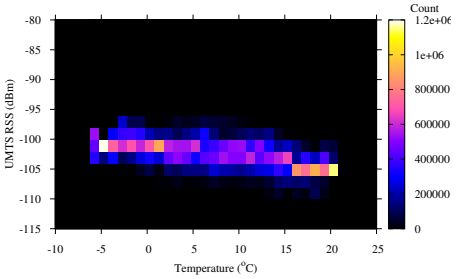
(a) Non-normalised Pressure



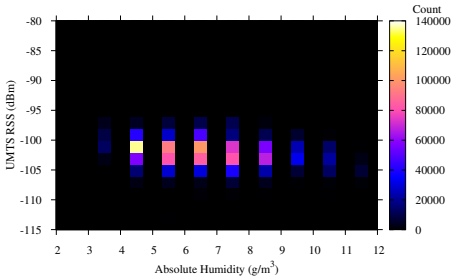
(b) Normalised Pressure



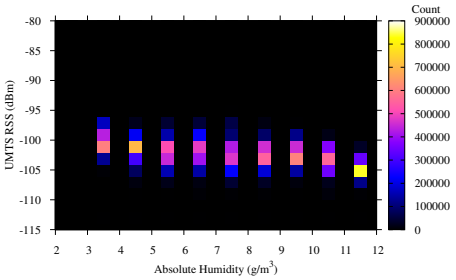
(c) Non-normalised Temperature



(d) Normalised Temperature

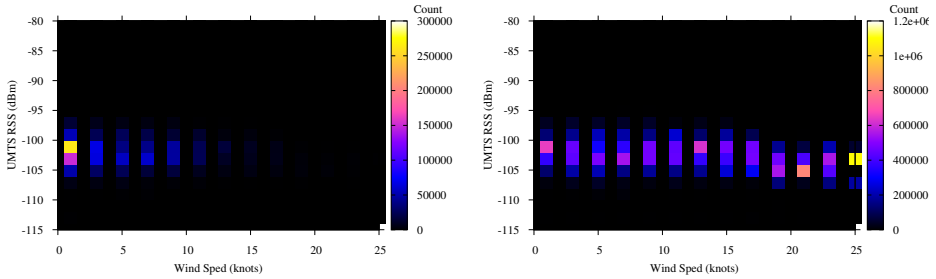


(e) Non-normalised Absolute Humidity



(f) Normalised Absolute Humidity

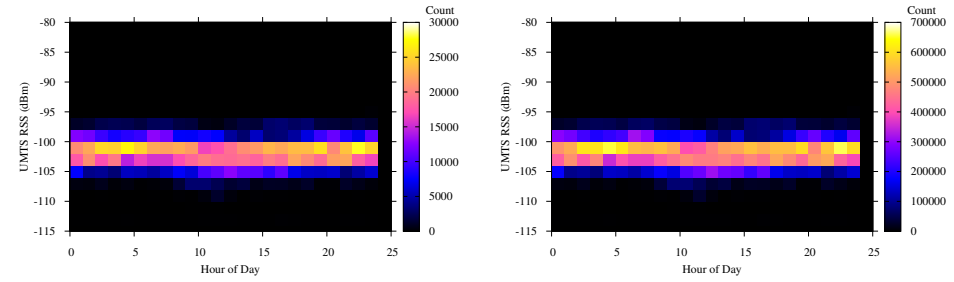
Figure 3.3: Meteorological effects on UMTS RSS.



(a) Non-normalised Wind Speed

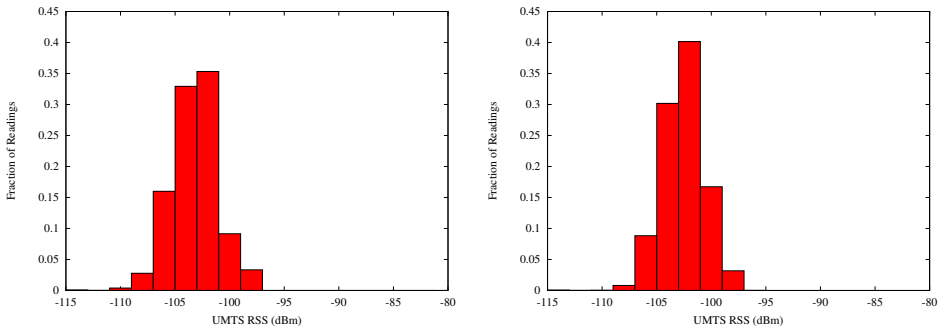
(b) Normalised Wind Speed

Figure 3.4: Effect of wind speed on UMTS RSS.



(a) Non-normalised Hour of Day

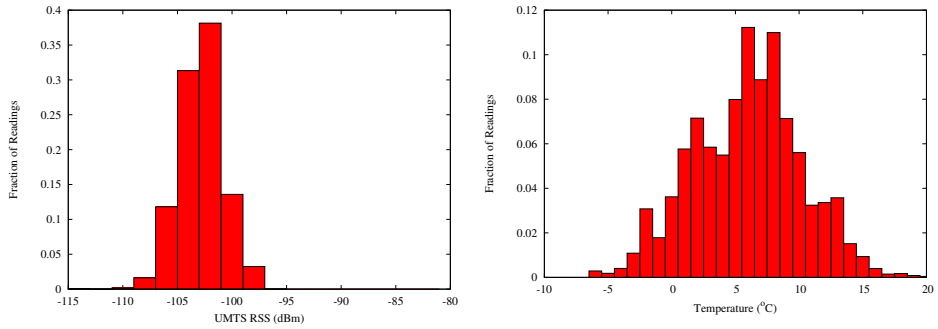
(b) Normalised Hour of Day



(c) Distribution of RSS During Working Hours (09:00 to 19:00 GMT)

(d) Distribution of RSS During Non-Working Hours (19:01 to 08:59 GMT)

Figure 3.5: Temporal effects on UMTS RSS.



(a) RSS Values (b) Temperature Values

Figure 3.6: Distributions of all recorded UMTS RSS and temperature values.

Figure 3.7: UMTS RSS around the city of Cambridge, UK. Data collected over 3 years. Darker colours represent higher RSS values, whilst green circles correspond to locations of UMTS base stations.

3.2.3 IEEE 802.11b/g Variability

In a similar vein, in order to evaluate whether IEEE 802.11 is affected by meteorological parameters, data from the Sentient Vehicles project was used concerning the RSS of a particular access point sited within the University of Cambridge Computer Laboratory. The investigation was confined to 802.11b, as the measurement hardware available was not capable of receiving 802.11g signals. Hence, the modulation scheme used was BPSK over DSSS, rather than BPSK over OFDM as used in 802.11g.[9]

The access point was a Cisco Aironet 1200 Series model, using two 2.14 dBi antennas, transmitting at a total power of 100 mW on a frequency of 2.437 GHz (channel 6). Other access points operating on the same channel were present in the area, contributing to the noise level on the channel. The signal was measured using a Lucent Orinoco PCMCIA card, connected to an external antenna providing a gain of 7.8 dBi. Logging was only carried out sporadically over seven months, but this included day and night periods. Probe frames were transmitted once every 2 seconds, and the signal strength of the reply frames received from all access points in range was recorded. In total, 1.3 million data points were recorded for the access point under investigation.

The location of the vehicle being used to record the data was normally confined to one particular parking bay. However, the data also includes points when the vehicle might have been parked in an adjacent bay, with a high-sided vehicle between it and the Laboratory building. This factor is likely to account for the somewhat bi-modal nature of the distribution of RSS (shown best in Figures 3.10(c) and 3.10(d)), with the small peak at lower (more negative) RSS values corresponding to the times when the view of the Laboratory was partially obscured. In addition, the data also includes readings when the vehicle was moving away from the building where the AP is sited, hence a few low RSS readings are to be expected.

3.2.3.1 Meteorological Effects

The graphs shown in Figures 3.8 and 3.9 were generated by a process analogous to that described in Section 3.2.2. None of the graphs show a correlation between 802.11b RSS and any of the meteorological parameters investigated, a fact which is not surprising, given day-to-day experience of using WiFi networks. This result does, however, permit us to discount one plausible source of variation in 802.11b RSS.

[9] In these measurements, beacon frames from the AP were used to measure RSS. In IEEE 802.11b and 802.11g, such frames are always transmitted using BPSK, unless 802.11g is set not to run in compatibility mode.

3.2.3.2 Temporal Effects

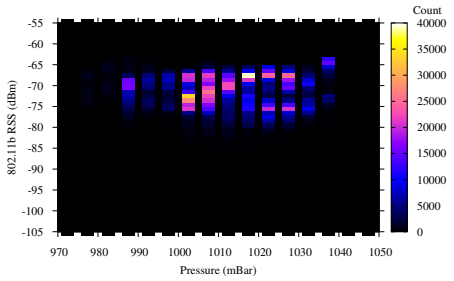
As regards the variation in RSS over time, Figures 3.10(c) and 3.10(d) show the distribution of RSS values during working (09:00 to 19:00 GMT) and non-working hours, respectively. These graphs appear to show that the spread of RSS values during non-working hours is somewhat less than at other times. This is borne out by Figures 3.10(a) and 3.10(b) which show that within working hours there are a lower number of light-coloured cells compared to outside these hours.

Figure 3.10 shows that 802.11b RSS does not remain at a fixed value over time. In this case, it appears that the signal power varies within the approximate range -66 to -76 dBm. Such variation can be ascribed to a variety of factors, in particular multipath effects that constantly change, and interference from other RF sources. The distribution of 802.11 RSS values recorded is approximately Gaussian, with a standard deviation of 3 dBm. This implies that 90% of the values lie within 6 dBm of the mean at any given location. Hence, if enough RSS data points are obtained for a given location, the approximate RSS at that location at any future time can be known.[10]

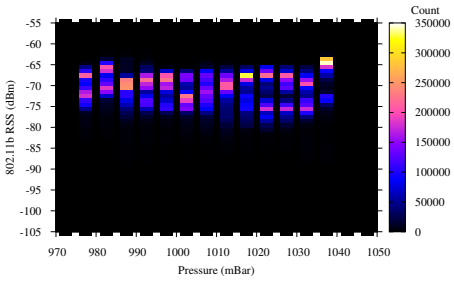
3.2.3.3 Applicability to 802.11g

Data were only collected for IEEE 802.11b, which uses a different transmission technique to that of 802.11g, although operates in the same frequency range. The OFDM transmission scheme used in 802.11g is theoretically more robust to multipath effects than 802.11b's DSSS. However, Bianchi *et al.* [89] found that 802.11g had a lower delivery success rate than 802.11b over 250 metre point-to-point links. Further investigation is therefore needed into the relative performances of 802.11b and 802.11g in outdoor environments. Therefore, the conclusions drawn above from data concerning 802.11b are not necessarily applicable to 802.11g. However, the work presented here provides an indication of how typical wireless local area network transmissions in this frequency band are affected (or, rather, are unaffected) by meteorological factors.

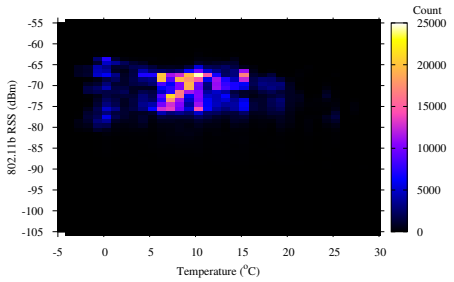
[10] Assuming that no radical changes to the environment, such as the obscuration of a transmitter by the construction of a new building.



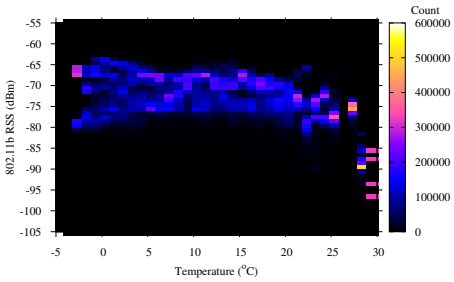
(a) Non-normalised Pressure



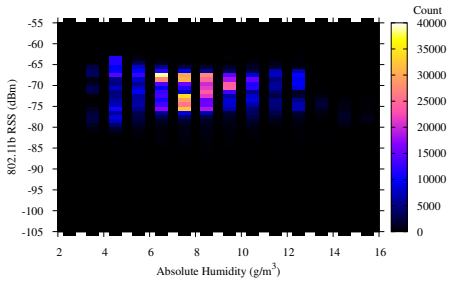
(b) Normalised Pressure



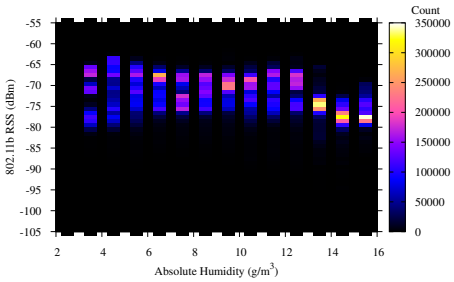
(c) Non-normalised Temperature



(d) Normalised Temperature

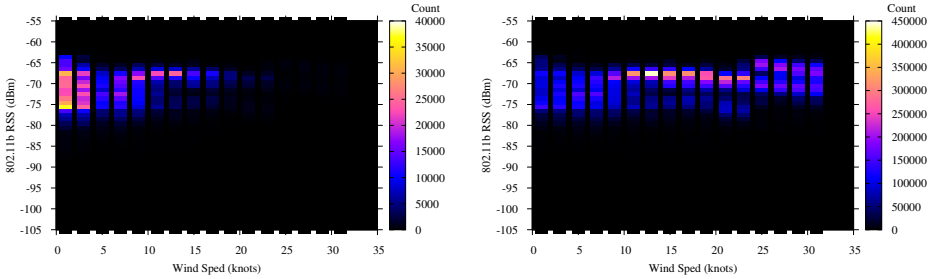


(e) Non-normalised Absolute Humidity



(f) Normalised Absolute Humidity

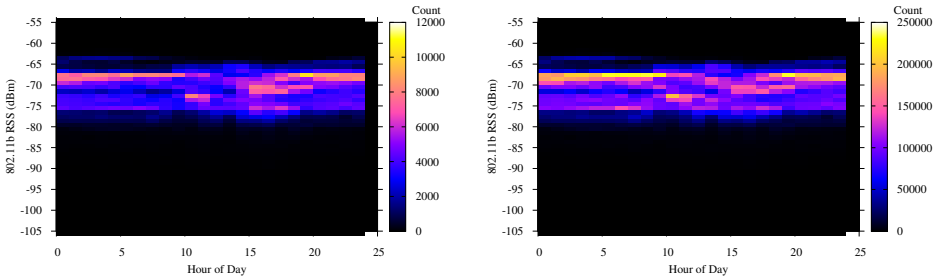
Figure 3.8: Meteorological effects on IEEE 802.11b RSS.



(a) Non-normalised Wind Speed

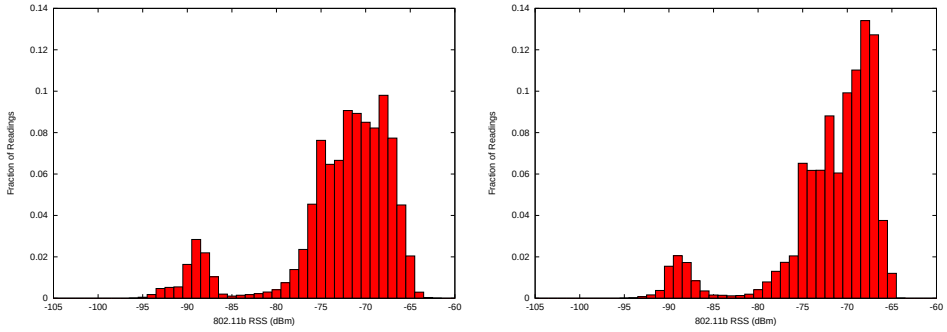
(b) Normalised Wind Speed

Figure 3.9: Effect of wind speed on IEEE 802.11b RSS.



(a) Non-normalised Hour of Day

(b) Normalised Hour of Day



(c) Distribution of RSS During Working Hours

(d) Distribution of RSS During Non-Working Hours

Figure 3.10: Temporal effects on IEEE 802.11b RSS.

3.3 UMTS Throughputs

Previous work in evaluating the performance of HSDPA over UMTS has been carried out, both with static and mobile nodes. Derksen *et al.* showed how throughputs achieved varied significantly in areas of low coverage [58]. Meanwhile, Diaz *et al.* [61] evaluated how video streamed to vehicles over UMTS performed, finding that the 95% confidence interval of delays was 120 ms, but the 99% interval was 800 ms, suggesting a wide spread.

Given that UMTS is designed for mobility, and has been the subject of previous evaluation, it was deemed unnecessary to evaluate it further in depth. However, to ensure realistic throughput figures were used for the remainder of the work described in this dissertation, qualitative experiments were carried out. These consisted of measuring the mean TCP throughput achievable using the Linux `wget` utility to download a 1 MB file from the laboratory web server. A test was carried out in five areas that had different values of UMTS RSS. The results are shown in Table 3.1.

UMTS RSS (dBm)	TCP Throughput (Mbit/s)
-63	1.28
-75	1.20
-93	1.08
-111	0.32
<-111	0

Table 3.1: Measured values of UMTS RSS & TCP throughput.

3.4 IEEE 802.11a Indoor Throughputs

Having established how IEEE 802.11b/g are affected by meteorological factors, it is useful to examine how other factors affect wireless technologies that are used for local area networks. A significant quantity of work exists concerning WaveLAN and 802.11b/g [68, 250, 241], but little on 802.11a. Given that the latter forms the basis for the 802.11p standard dedicated to vehicles, it was considered beneficial to investigate this technology, rather than repeat previous work.

3.4.1 Motivation for Indoor Experiments

Link quality in wireless deployments is known to be affected by a number of factors, including antenna height, radio shadowing, variation in RF environment, and receiver sensitivity [82]. Transmissions to/from vehicles travelling through urban

canyons suffer in particular from multipath effects, where the signal travels a variety of different paths in parallel, resulting in interference at the receiver. Such multiple paths are in the main due to the presence of many reflective surfaces, particularly tall buildings on both sides of a street. This environment is very similar to the wave guide formed by an indoor office corridor.

Several caveats, however, apply. In particular, the width of a corridor versus an inner city road is very much smaller, resulting in the difference in path lengths between reflected and direct transmissions being much smaller. This in turn will mean that the interference patterns are likely to be less harmful to the throughput achievable in a corridor as compared to that in an urban canyon[11]. Also, the objects that move in a corridor tend to be humans (well suited to absorbing signals in this frequency band), rather than (metallic) vehicles, that reflect the signal significantly. Despite these shortcomings, the office corridor scenario offered a more controlled experimental environment than an outdoor area. It also facilitated the carrying out of long-term tests without security or bureaucracy concerns.

Experiments were performed to assess the effects of movement (varying the propagation characteristics of the environment), access point positioning, and access point beacon interval on the throughputs achievable using 802.11a. As a further simplification, static terminals were used, in order to isolate any effects to solely those parameters that were being varied (e.g. AP position). After these controlled *indoor* experiments, measurements of the performance *outdoors* with a moving vehicle and an otherwise stationary environment, were carried out. These are described in Section 3.5.

3.4.2 Experimental Set-up

3.4.2.1 Environment & Equipment

Linux-based computers were used as the source and sink for all tests. One machine was connected using 100 Mbit/s wired Ethernet to a Cisco 1200 series wireless AP which had an 802.11a module installed, transmitting on a frequency of 5.32 GHz. The other computer used an NEC Warpstar 802.11a/b/g PCMCIA interface card (with no external antenna), and ran Linux kernel version 2.6.11.4.

For tests examining indoor AP positioning (Section 3.4.3), the set-up was as shown in Figure 3.11. The AP was placed at one end of a 46 metre long, 2.3 metre-wide office corridor, shown in Figure 3.12. The source laptop was placed on the floor at the other end of the corridor. The AP was situated offset from the centreline of the corridor, 0.55 metres from one wall, whilst the source laptop was placed on the centreline. The only obstructions in the corridor were on the same side as

[11]The reasoning for this is that the smaller the difference in arrival time, the less likely each transmitted symbol is to overlap with another transmitted before/after it. See Section 2.3.1 for further details.

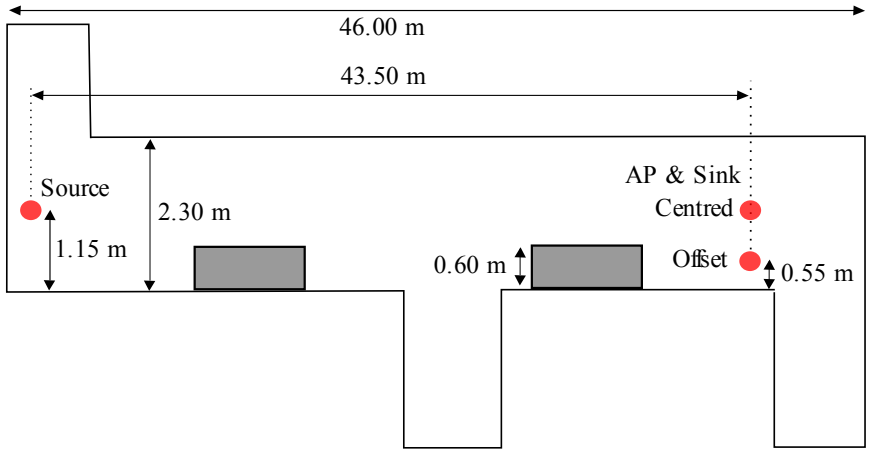


Figure 3.11: Experimental set-up for corridor-based experiments. Significant obstructions in the corridor are marked by the two grey rectangles.

the access point, and extended 0.6 metres into the corridor. The AP therefore did not have line of sight to the laptop in its offset position. The AP's antenna, (an omnidirectional Cisco AIR-RM21A with a gain of 5 dBi, shown in Figure 3.13) had its largest face oriented parallel to the centreline of the corridor.

3.4.2.2 Throughput Measurements

All throughput tests were performed using the iperf[12] network measurement tool, with the source being the laptop with the 802.11a PCMCIA card, and the sink being a workstation connected to the AP over wired ethernet. In each test UDP datagrams of 1470 bytes in size were transmitted from the source to the sink. The average throughput (of uncorrupted packets received by the sink) for each 0.5 second interval was calculated. The offered load (i.e., the rate at which the source sent packets to the access point) was 30 Mbits/s. The direction of transmission was deliberately chosen so that the sender of UDP packets would do so via its wireless interface, rather than through a wired interface to the AP. This ensured that buffers associated with the wireless hardware influenced the rate of transmission directly, rather than the wired link causing overflow at the AP's buffers. In order to saturate the wireless link, the offered load was selected to be a little greater than the highest throughput ever achieved. Thus, there was almost always a small amount of packet loss even when the link was performing under optimal conditions.

[12] `http://dast.nlanr.net/Projects/Iperf/`

Figure 3.12: The office corridor used for 802.11a experiments, looking from the location of the AP.

Figure 3.13: The Cisco 1200 series AP with omnidirectional patch antenna used for 802.11a experiments.

Whilst TCP is commonly used for many data transfers, UDP was chosen for these experiments. The reasons for this were:

- TCP infers congestion from packet loss and scales back its send window accordingly. For these experiments, the aim was to ascertain how throughput was affected by the wireless propagation characteristics of the channel, rather than due to protocol-induced throughput reductions.
- TCP uses an ARQ mechanism to retransmit lost packets, causing the sending of normal data to cease temporarily. Again, the desire was to analyse the performance of the underlying data-link layer rather than ARQ-based protocols being carried over it.

3.4.3 Access Point Placement

In deploying wireless networks it is important to consider the placement of the access point in order to maximise its coverage area, but also to maximise the throughput achievable by its clients. This is particularly the case for both indoor corridors and urban canyons, which can act as wave guides for wireless signals, and in which there is a high degree of movement. The aim of the experiments described in this Section was to provide an indication of the performance impact (if any) of APs if deployed on posts to the side of roads (which may well not have line of sight very far down the road due to buildings or foliage), rather than on gantries centred above them. The costs of gantries (both in terms of installation and maintenance, as road closures are required) is significant, hence this question has appreciable economic impact. Hence, the cases of a centreline and an offset AP were compared in a controlled indoor environment, in order to better understand the effects placement has on the overall performance.

3.4.3.1 Issues With Current Approaches

Access point placement is an everyday problem, the objectives being to maximise coverage whilst minimising inter-AP interference (or overlap, as this implies wasted resources). In order to calculate where APs should be placed, their coverage must be predicted, which in turn requires knowledge of how the radio signals from them propagate. In the field of radio engineering, predicting propagation and coverage is a well researched, but notoriously difficult problem domain. Prediction using simple path models [182] and dominant paths [244] are two approaches specifically targeted at indoor situations. Simple path models incorporate limited numbers of reflections. Dominant path models compute losses for multiple paths, as in ray tracing, but only the most significant one is chosen as “dominant” and propagated to the next step of the simulation. Performance varies, with the latter

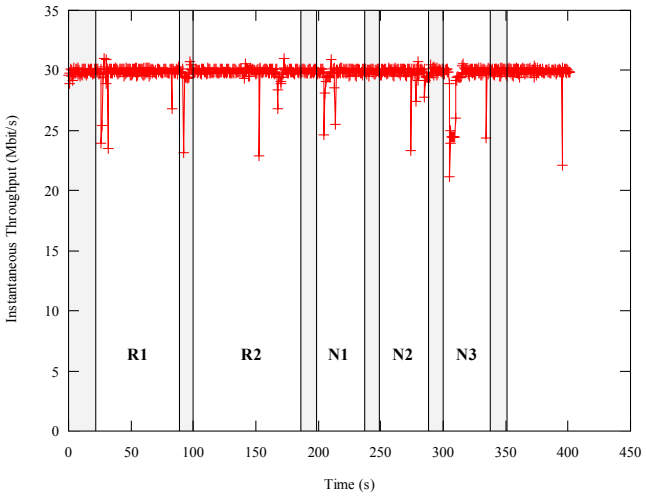


Figure 3.14: Throughput down the corridor with the AP situated on the centreline of the corridor.

approach requiring a detailed database of the locations of walls and furniture in order to be accurate [245].

Using such propagation models, optimisation algorithms can then be applied to find where APs should be placed, two examples being genetic algorithms [156] and neighbourhood search [128]. However, the majority of work in this area has been simulation (without real-world validation) rather than experimental deployment. In contrast, experimental work has shown that even in homes, coverage is not as might be expected [183]. Asymmetric connectivity between rooms can occur, and coverage black spots are present even in the areas that were relatively close to an access point. Hence, further experimental work is required to establish how wireless local area network technology performs in real indoor environments.

3.4.3.2 Experimental Method

As described in Section 3.4.2, the set-up consisted of a computer connected to an AP at one end of a 46 metre long corridor, and a laptop (the source) at the other. Experiments were carried out to ascertain how a person walking along the corridor in different ways affected the throughput that was achieved.

Figures 3.14 and 3.15 show the throughput achieved by the IEEE 802.11a protocol as the experiments were carried out. Regions in grey are *stabilisation* periods, where no movement took place in the corridor. These were of a minimum of 10 seconds in duration, and were to ensure that a steady state was reached before each new test began. Each trace is divided by these stabilisation periods into the following intervals:

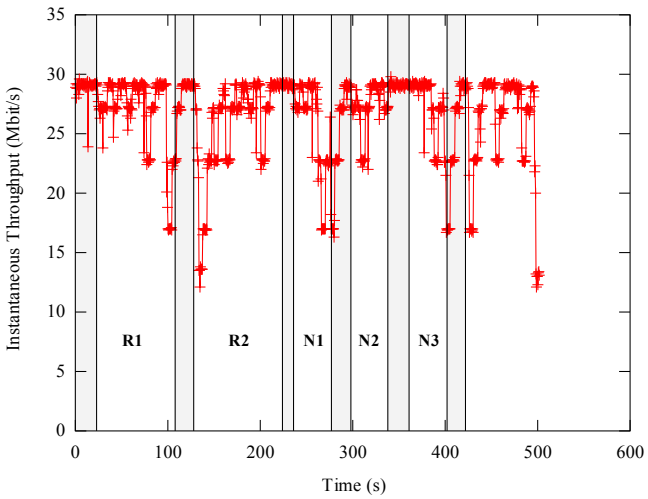


Figure 3.15: Throughput down the corridor with the AP offset from the centreline of the corridor.

- **Random walk 1 (R1):** one person walked down the corridor from the AP to the source, along a random path, opening and closing doors, and making brief stops.

- **Random walk 2 (R2):** similar to the first random walk, but in the opposite direction. In addition a large door behind the AP was opened and closed.

- **Normal walk 1 (N1):** one person walked from the AP to the source at an approximately constant speed, down the centreline of the corridor.

- **Normal walk 2 (N2):** similar to normal walk 1, but in the reverse direction.

- **Normal walk 3 (N3):** identical to normal walk 1.

The final portion of the trace after the last stabilisation period corresponds to when the equipment was still running after the experiments had finished, but before data collection was stopped, and can be ignored.

It should be noted that the random walks, by their very nature of not being straight lines, took longer to execute than the normal walks, and hence take up wider portions of the traces. Moreover, the path taken for each random walk and the actions performed over it were very similar on each of the tests carried out, but the timings and speeds were not quantitatively measured, and hence cannot be said to have been identical.

3.4.3.3 Centred Versus Offset

As can be seen from Figures 3.14 and 3.15, the variation in throughput for the offset access point is much greater than that for the centred one. This is because for the centreline case the transmitter (the source) and receiver (the sink) have line of sight, and hence the signal travels along the direct path between the two. In contrast, with an offset access point, there is no line of sight (NLOS). Here, changes in the environment, such as doors opening, cause the paths of reflected waves to change significantly.

It is interesting to note that much of the variation for the AP on the centreline (Figure 3.14) occurs at the point where the experimenter was near the transmitter or receiver, and hardly any variation is seen when they were between the two. This effect is due to more of the direct paths between the transmitter/receiver being blocked as the experimenter approaches one or the other. At the midpoint of the corridor, the angles that the signal must traverse in order to not be blocked by the experimenter are not large, and throughput is higher. Similar, but more pronounced effects are seen in the NLOS case depicted in Figure 3.15. This increased impact is due to the communication relying entirely on transmission paths that are *not* direct. Hence, the signals' paths will need to include even more reflections in order that it can turn through the necessary angles to reach the receiver.

Overall, it can be concluded that in environments where there are wave-guides, such as corridors in buildings, or (bearing in mind the caveats mentioned in Section 3.4.1) urban canyons, there is significant advantage in placing the access point centrally, rather than offset. This implies that placing an access point on a ceiling (centrally) has more benefits than simply the height. It may also mean that APs on gantries provide better coverage down a street than those attached to buildings. This implies that there is a trade-off between the increased cost of deployment of a centred AP and the benefit of a lower variation in throughput that it can result in.

3.4.4 Beacon Interval

The IEEE 802.11 standard specifies that in any type of 802.11 network, beacon frames should be periodically transmitted, allowing the network to be discovered by mobile nodes. Beacons contain information such as the network name, the rates supported, and other radio transmission parameters. They can be used to ensure that stations hibernate and awake from hibernation at the correct times, thus saving power. They are also important in maintaining a synchronised clock over all stations to within 4 μs, and for conveying frequency-hopping sequence information (for legacy FHSS equipment), or current channel number for DSSS equipment. From a mobile user's perspective, a shorter beacon interval allows faster identification of a network, and hence more seamless mobility. However, more frequent beacons mean that the channel is free less often for data packets.

The interval between beacon frames is normally 100 Kμs[13], but can be varied between 20 and 4000 Kμs on Cisco 1200 series access points. Experiments were performed examining how varying this interval affects the throughput achieved at various ranges, in order to provide guidance on what the optimal interval for a particular environment is.

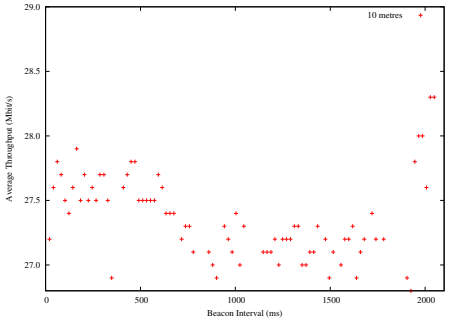
3.4.4.1 Previous Work

A variety of simulation and analytical work [190, 217] has been carried out on optimising the 802.11a protocol by varying parameters such as fragmentation threshold, modulation scheme, or utilising the Point Co-ordination Function (PCF) instead of the Distributed Co-ordination Function (DCF). The effect of beacon interval on network discovery time has also been examined [230]. Here, the authors used ns-2 simulations to evaluate how small the beacon interval could be made without impacting throughputs, as smaller intervals decreased discovery times. Meanwhile, a method of adapting the beacon interval depending on network load has also been patented [227]. However, to the best of the author's knowledge, no work has until now been carried out examining how beacon interval affects throughput, or how this effect varies with transmission range.

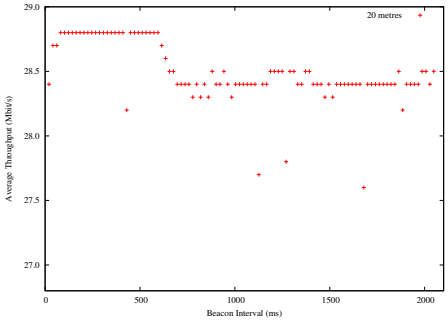
3.4.4.2 Experimental Method

In order to assess how the average throughput varied with beacon interval, experiments were carried out in the office corridor described in Section 3.4.2. The source laptop was placed on a trolley, approximately 1 metre above the floor, and was moved to different positions in order to vary the distance between the transmitter and receiver (the *transmission range*). The AP was placed on the centreline of the corridor. Each different value of the beacon interval was tested with an iperf connection lasting 300 seconds. Beacon intervals were varied between 20 and 2000 Kμs, in steps of 20 Kμs (i.e. between 20.48 and 2048 ms, in steps of 20.48 ms). The average UDP throughput as reported by iperf was recorded for each 0.5 second interval, with the entire process being automated in order to allow large numbers of readings to be collected. All beacon intervals were tested at transmission ranges of 46, 40, 30, 20, and 10 metres. All experiments were carried out overnight when the propagation environment is assumed to have been predominantly static, given that (in general) no one was present in the offices outside the hours of 08:00 to 20:00, or if they were, there was almost zero activity in the corridor. Each test lasted 300 seconds, resulting in 600 throughput readings, which were then used to generate a mean throughput for each beacon interval at each transmission range.

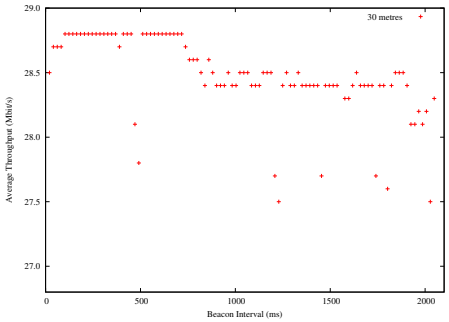
[13]This corresponds to units of 1024 μseconds, these being the units the AP utilises. Hereafter these quantities are converted to milliseconds.



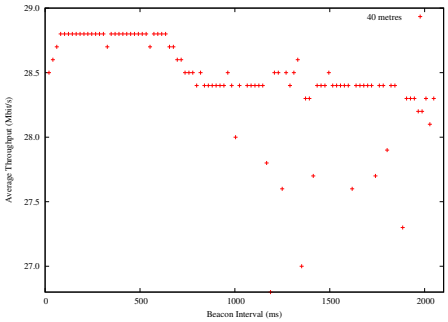
(a) 10 metres



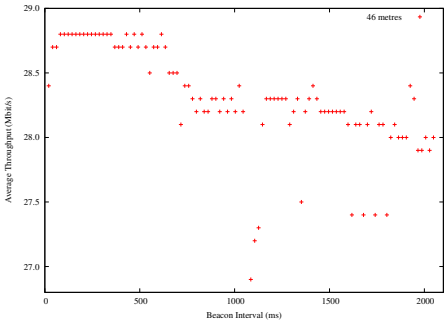
(b) 20 metres



(c) 30 metres

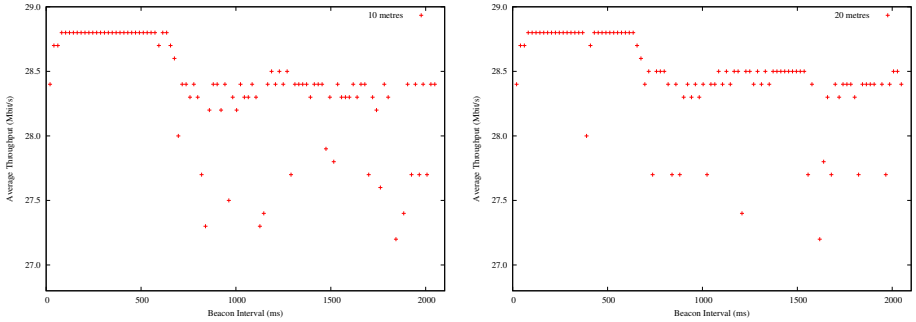


(d) 40 metres



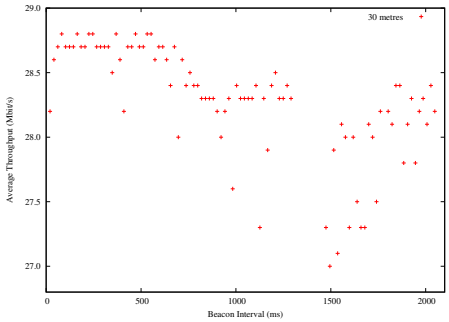
(e) 46 metres

Figure 3.16: Effect of beacon interval on average 802.11a throughput with varying transmission range with a static environment.

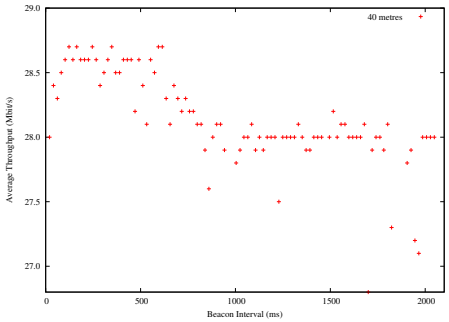


(a) 10 metres

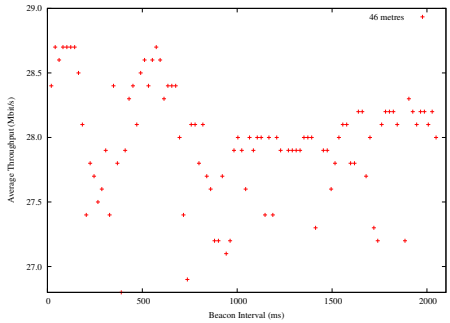
(b) 20 metres



(c) 30 metres

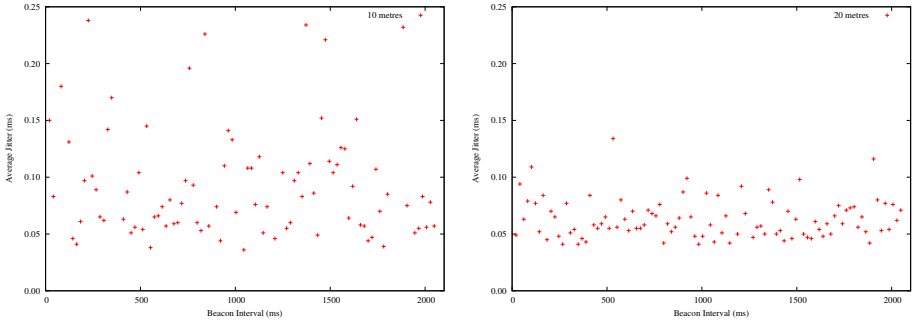


(d) 40 metres



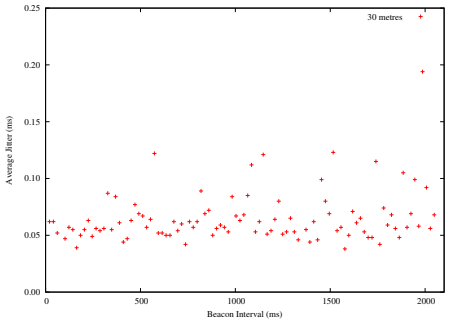
(e) 46 metres

Figure 3.17: Effect of beacon interval on average 802.11a throughput with varying transmission range with a dynamic environment.

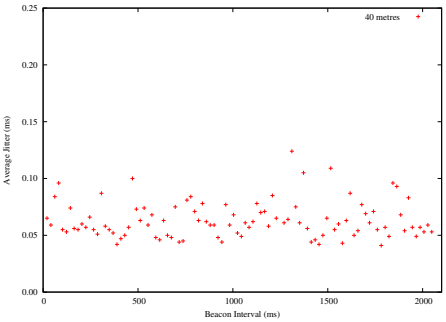


(a) 10 metres

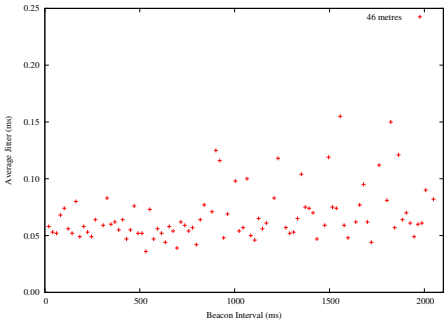
(b) 20 metres



(c) 30 metres

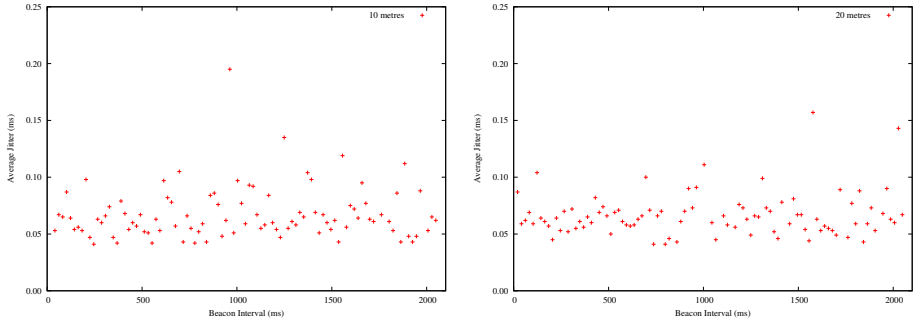


(d) 40 metres



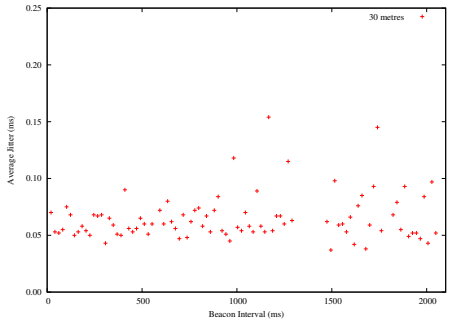
(e) 46 metres

Figure 3.18: Effect of beacon interval on average 802.11a jitter with varying transmission range with a static environment.

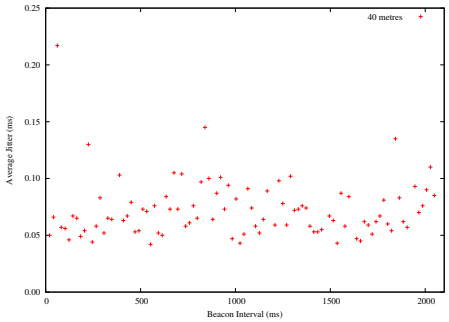


(a) 10 metres

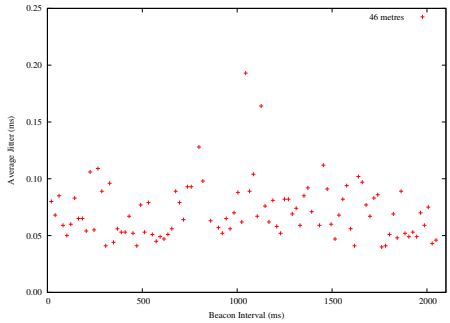
(b) 20 metres



(c) 30 metres

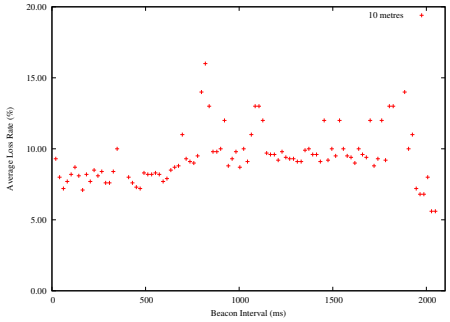


(d) 40 metres

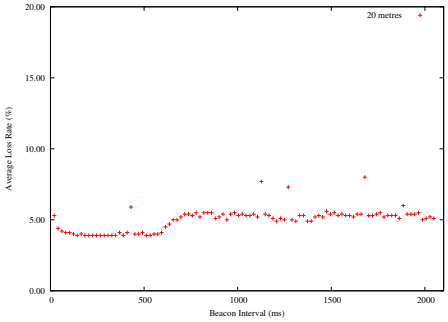


(e) 46 metres

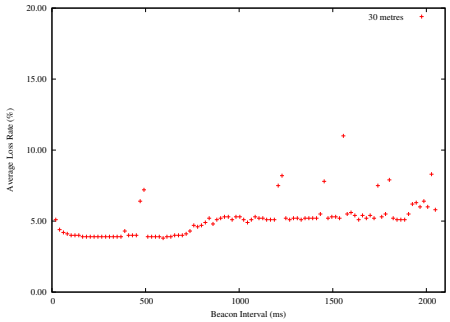
Figure 3.19: Effect of beacon interval on average 802.11a jitter with varying transmission range with a dynamic environment.



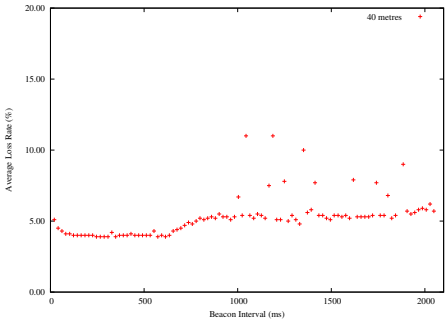
(a) 10 metres



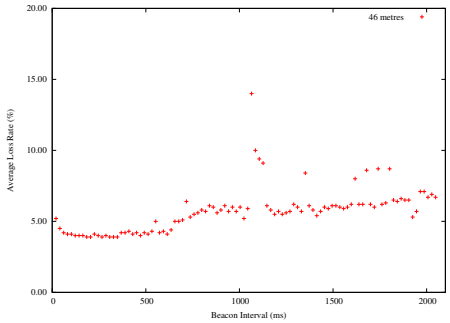
(b) 20 metres



(c) 30 metres

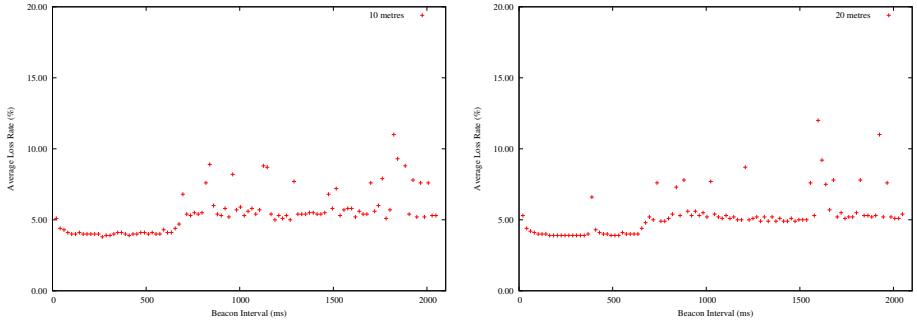


(d) 40 metres



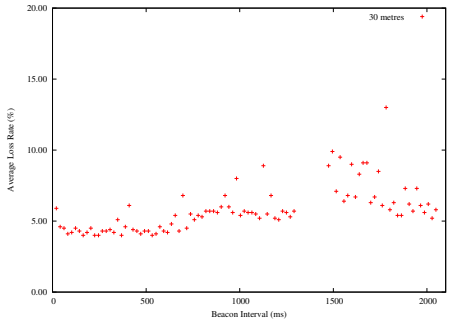
(e) 46 metres

Figure 3.20: Effect of beacon interval on average 802.11a loss percentage with varying transmission range with a static environment.

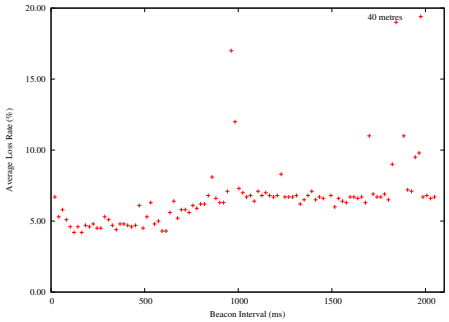


(a) 10 metres

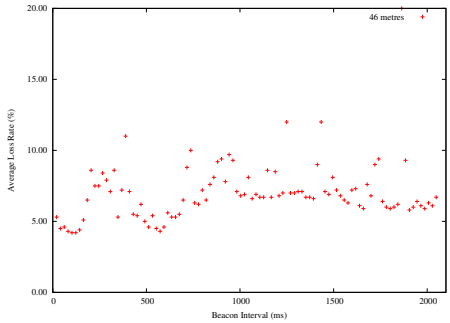
(b) 20 metres



(c) 30 metres



(d) 40 metres



(e) 46 metres

Figure 3.21: Effect of beacon interval on average 802.11a loss percentage with varying transmission range with a dynamic environment.

To confirm that these results were not due to a single product's implementation of the standard, tests were also carried out using an Intel-based IP2200 chipset, in an entirely different laptop running Fedora Core 4 Linux (kernel 2.4.17), and using the same Cisco AP. The results showed the same patterns for both hardware configurations. The original AP was then substituted with a Cisco 1130AG and some of the tests were repeated with the Intel chipset. The results showed the same general shape as before, although with much greater variability in the traces.

3.4.4.3 Effect of Distance

Figure 3.16 shows how the mean throughput for each beacon interval followed a similar trend across all transmission ranges, except for the 10 metre range. Overall, UDP throughput decreases with increasing beacon interval (i.e. fewer beacons transmitted per second), but the relationship has two distinct plateaux. Separately, as the transmission range is increased, the throughput achieved at higher beacon intervals begins to fall. At 20 metres, the second plateau extends to the right-hand edge of the graph, whilst at 30 and 40 metres there is a clear drop in throughput above an interval of 1800 ms. At a range of 46 metres, the interval at which the decrease begins is reduced to 1500 ms. Thus, for a given throughput a lower beacon interval is necessary as the transmission range is increased. From Figure 3.16, it appears that the optimal beacon rate in terms of UDP throughput for most transmission ranges (in static environments) is between 200 and 400 ms. Interestingly, the optimal *range* appears to be 20 metres, this being the one that has the smoothest trace in Figure 3.16.

The low performance at a range of 10 metres could be due to the receiver circuit being overloaded by its proximity to the transmitter. This results in receiver amplifier non-linearity, such as clipping. Such behaviour distorts the amplitude of the received signal, and hence for higher order modulation schemes running over OFDM results in a significant reduction in throughput. However, it is not clear why clipping should not also have been present for the dynamic environment experiments: the only possible explanation could be movement of people in the corridor absorbed enough signal energy to prevent clipping, whilst not impacting throughput. To ensure that the effect was not due to excessive RF noise on the channel, the background noise power levels at that frequency were measured for 10 minutes. No sources of interference were detected. A second possible explanation is that the receiver at the 10 metre range happened to be located in a null zone, causing greater packet losses. This would also explain why the poor performance was not seen with the dynamic environment, as the equipment might have been set up at a location that was a few millimetres different to that used for the static experiments. This would have potentially been enough for the antenna to not be in the null zone.

Higher beacon intervals (> 920 ms) show more variability in throughput than lower intervals. This is a consequence of fewer beacons being received per second at

higher intervals, and hence a greater period of time elapsing between a single beacon being lost, and another being received. Section 3.4.4.5 considers this issue.

Beacon intervals of less than 82 ms exhibit a reduced overall throughput because of the time the channel is occupied transmitting them. The simulations in [230] reported this point as being at 60 ms, thus there is a good match with a previous work. Each beacon transmitted by the Cisco 1200 AP used for these tests was 301 bytes long. Transmitted with the lowest modulation scheme (BPSK) with a throughput of 6 Mbit/s, each beacon would take 0.05 ms to transmit. If a beacon interval of 20 ms is used, this equates to 2.50 ms of each second being used for beacon transmissions. For an interval of 200 ms, this decreases to 0.25 ms. This difference of 2.25 ms in the quantity of time dedicated to beacon frames, if it could be used with a UDP throughput of approximately 30 Mbit/s, would result in an extra 0.06 Mbit/s throughput; comparable to that achievable with an analogue telephone line modem. Figures 3.16 and 3.17 show that the differences can be even more pronounced, with the difference in throughput between 20 and 200 ms for dynamic environments being 0.38 Mbit/s. This is likely to be due to other effects such as a beacon transmission being proceeded by a mandatory interframe spacing (DIFS) period (0.034 ms for 802.11a, as given in Table 93 of [115]), before another station can use the channel.

As regards jitter[14], Figures 3.18 and 3.19 do not show any correlation between transmission distance and the degree of jitter. However, it does appears that with longer beacon intervals jitter is more likely to be high, with the distribution of points on each graph more scattered along the vertical axis as beacon interval increases. This effect is due to the greater number of link-layer retransmissions required (on average) for each UDP packet to arrive. Higher UDP loss rates, particularly at higher transmission ranges as shown in Figure 3.20, imply that for those packets that do arrive link-layer retransmissions are likely to have occurred. Each retransmission increases the time taken for the UDP packet to be noted as having arrived, compared to those that arrive on initial transmission. Thus the variation in the time taken for packets to traverse the link is large. As would be expected, the percentage of UDP packets lost (Figure 3.20) shows a similar (but inverse) "knee" as that for UDP throughput (Figure 3.16) at a beacon interval of approximately 630 ms.

3.4.4.4 Static & Dynamic Environments

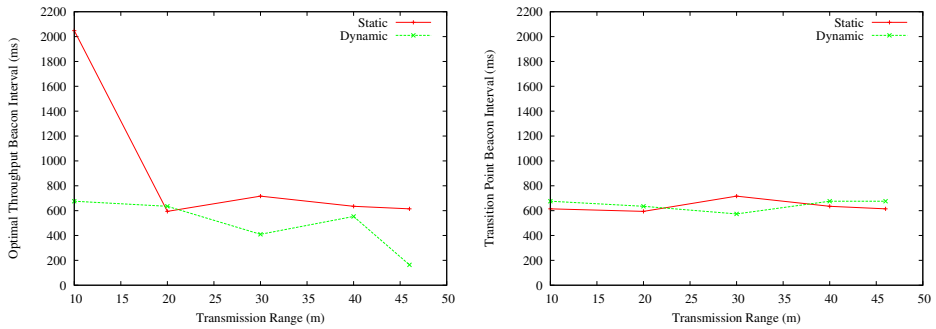
As discussed in Section 2.3.3, the throughput of a stream carried over a wireless link is dependent on the magnitude of the transmitter's signal strength experienced at the receiver, which in turn is dictated by how RF propagates in that particular environment. To assess how beacon interval and the degree of fast fading combine to affect throughput, experiments were performed in the office corridor described in

[14]The jitter calculation was as defined in RFC 3550 [199], Section 6.4.1.

Section 3.4.2 during working hours and overnight. The results are therefore shown for *dynamic* (office hours), and *static* (overnight, when it is assumed that there was little or no movement in the corridor) environments. A dynamic scenario presents a more challenging RF environment because the paths taken by radio waves from the transmitter to the receiver will vary. For example, if a door is closed in the corridor, the signal may be reflected off it where previously it did not. Such reflections may constructively or destructively interfere with one another, thus changing the quality of the connection at the receiver. Similar effects will take place in an urban canyon, due to the movement of vehicles. In such a situation, both the transmitter/receiver in a vehicle is likely to be moving *in addition* to the motion of surrounding traffic. However, by comparing an environment where movement was taking place but where the communicating nodes were *stationary*, to a completely static environment, it was possible to ascertain the significance of the motion of objects in the environment, rather than the motion of the communicating nodes themselves.

For static environments (ignoring the 10 metre range readings, as discussed in Section 3.4.4.3), the optimal throughput (the highest peaks on the graphs in Figures 3.16 and 3.17), can be achieved with intervals of at least 600 ms. The 10 metre transmission range is different, as the maximum throughput occurs at an interval of 2048 ms. For dynamic environments the optimal beacon interval reduces with increasing transmission range to 164 ms at 46 metres. The maximum throughput achieved in both cases is approximately 28.8 Mbit/s. The marked difference in interval is explained by the more challenging dynamic propagation environment, where beacon frames are more frequently lost, thus requiring a smaller beacon interval to achieve the same number of beacons received in a given period of time. In contrast, in a static environment the channel does not vary so significantly, and therefore fewer beacon frames per second are required. Meanwhile, the data in Figure 3.16 shows less variation between the different transmission ranges compared to that in Figure 3.17, further indicating that receivers in dynamic environments encounter more variable throughputs.

Figure 3.22 shows how the transmission range influenced the beacon interval at which the maximum throughput was achieved. Figure 3.22(b) depicts at which beacon interval the "knee" in the throughput graphs in Figures 3.16 and 3.17 began (i.e. the lowest interval at which throughput began a steep curve downwards). As described above, the beacon interval required to achieve maximum throughput decreases with increasing transmission range for both static and dynamic environments. Hence, the maximum throughput at any range (excepting 20 metres) required a lower interval than that for the corresponding static experiment. The point of transition from the plateaux on the graphs to the "knees", shown in Figure 3.22(b) does not show any trend, with the curves for both dynamic and static scenarios almost flat. On this graph too, dynamic environments always (except at a range of 30 metres) have a higher beacon interval of transition than for the corresponding static experiment. This is due to the increased interference caused by movement in dynamic environments.



(a) Optimal throughput

(b) Transition point

Figure 3.22: Throughput-optimal and transition point beacon intervals at different ranges for static and dynamic environments.

Jitter is broadly similar in static and dynamic environments (Figures 3.18 and 3.19). This is somewhat surprising given that greater loss rates would be expected to imply both lower throughputs and a greater variation in the times taken for packets to traverse the link (i.e. greater jitter). This may be because (particularly at long transmission ranges) the losses in dynamic environments are so great that all retransmissions for many packets fail. This is borne out by the UDP loss rates shown in Figure 3.21, which are higher for dynamic environments for the reasons discussed above. With such high loss rates, the packets which do not reach the receiver cannot have their traversal times measured, and hence the variation in traversal time (jitter) is deceptively low.

3.4.4.5 Explaining The Effect of Beacon Interval

In order to ascertain why modifying the beacon interval should affect throughput, the same static environment experiment was carried out once more. Beacon intervals of 20, 200, and 1000 ms were used, this time in conjunction with another laptop acting as a sniffer. The sniffer recorded a trace of which frames were transmitted over 30 seconds in the middle of each 300 second run. At a beacon interval of 20 ms, more than 1,000 beacon frames were captured, indicating that a significant amount of throughput was being utilised for beacon transmissions. At 200 ms, fewer beacon frames were captured, whilst throughput was at its maximum. At 1000 ms, regular (approximately once per second) MAC-layer re-association requests by the laptop to the AP were observed. It therefore appears that above a certain threshold beacon interval, the end station assumes that its association with the AP has been revoked. This effect was observed with two different models of AP. Hence, whilst increasing beacon interval increases throughput up to a point, above this threshold, throughput decreases due to the once per second re-association pro-

cedure that a client must perform. Therefore, in dynamic environments (or with longer transmission ranges), where environmental effects cause many beacons to be lost, the client's *perceived* interval between AP beacons can be long, causing re-associations to be necessary.

3.4.5 Impact of Beacon Interval on Network Design

Modifying the beacon interval of an AP will affect the time taken by nodes to discover it. If the interval is very large, then this may cause problems when stations move quickly and do not discover the AP in time to perform a seamless handover. In addition, these experiments showed that at long transmission ranges or in dynamic environments, the usage of long beacon intervals causes stations to constantly re-associate, thus harming transport protocol throughputs. Network designers must therefore carefully choose the interval based on the likely use of the AP. For example, in a vehicular context, a very high beacon rate may be more desirable (in order to aid discovery) than the extra throughput that would otherwise be available if a moderately longer interval were used.

3.5 IEEE 802.11a Outdoor Throughputs

3.5.1 Introduction

Section 3.4 examined several of the factors that affect the performance of IEEE 802.11a in a controlled indoor environment. This Section considers how the technology performs in a more realistic scenario, namely to send data to a moving vehicle. In particular, the effect of *low*, rather than high, vehicular speeds on throughput is examined. This is important given that the greatest usage of such technology is likely to be in urban areas where prevailing traffic speeds are low.

3.5.2 Related Work

Recent years have seen an explosion in the quantity of research into how wireless local area network communications perform when used in vehicular contexts. Much of this work has centred around 802.11b and 802.11g, beginning in 2002 when Singh *et al*. examined how 802.11b performed in urban, sub-urban and freeway environments [210]. Notably, they concluded that throughput decreased with increasing speed. In addition, separation distance between the communicating vehicles dictated the optimal packet size that should be used for the connection, with smaller sizes being better for larger separations, which incurred greater packet loss rates.

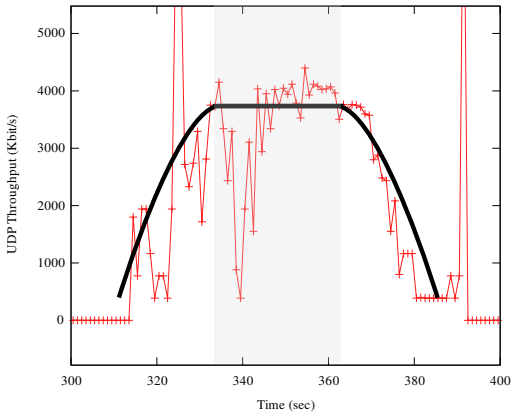


Figure 3.23: Connection phases of a vehicle passing an 802.11b access point: the black line divides the three phases of entry, production, and exit. The production phase is shaded in grey.

In 2003, Bergamo *et al*. established that 802.11b communication was possible between two moving vehicles at a relative speed of 240 km/h [17]. However, they also showed that the presence of urban clutter caused significant numbers of packets to be lost, and jitter to increase markedly. This demonstration of successful transmissions at high vehicular speeds further ignited interest in using off-the-shelf 802.11 technology for intervehicular applications.

Ott and Kutscher began the Drive-thru Internet project in 2004, where they evaluated the performance of 802.11b in highway scenarios, and later 802.11g. For 802.11b, TCP throughputs of 4.5 to 5 Mbit/s were achieved, the coverage area of an access point being found to be approximately 200 metres in radius [179]. In the case of 802.11g, ranges of up to 1.25 km were possible, with TCP throughputs of 15 Mbit/s at speeds of 80 km/h, resulting in cumulative transfers of up to 110 MBytes [178]. Ott and Kutscher were the first authors to detail how a connectivity session by a vehicle could be divided into the entry, production, and exit phases, as shown in Figure 3.23. Some authors have subsequently pointed out that the production phase, whilst providing the highest throughputs, does nonetheless suffer from interruptions [155], as the results in Section 3.5.4 also show.

Later, Wu *et al*. performed experiments on a highway using 802.11b [247]. These showed that a consistently good connection could be achieved at a range of 150 metres, whilst sporadic connectivity was possible at up to 600 metres. Curves and bridges impacted the connectivity significantly, suggesting that line of sight is an important factor for this technology. Meanwhile, Gass, Scott and Diot carried out experiments concerning 802.11b in a desert scenario [85], i.e. where there was no RF interference, and no other moving objects that would alter the radio propagation environment. They assessed connectivity from a laptop inside the vehicle to a roadside AP at speeds of up to 120 km/h, concluding that such connectivity provided

useful transfer sizes, and was *not* affected by the speed of the vehicle.

Most recently, MIT's CarTel project carried out an evaluation of the performance that could be expected by vehicles when making use of 802.11b/g APs located inside private homes or businesses [24]. The research found that performance did *not* appear to be related to vehicle speed, and on average the coverage regions were of 96 metres in length (though the top 10% had up to 300 metres of coverage). Simultaneously, Wellens *et al*. carried out experiments on 802.11b and 802.11g, varying vehicle speed [239]. They concluded that speed did *not* affect the performance of these technologies, with speeds of 120 km/h yielding good performances.

A limited quantity of simulation work has been carried out concerning 802.11a, or its vehicular-specific derivative, 802.11p. This has mainly focussed on the priority mechanisms built into the protocol [222, 70], but physical layer evaluation has also been simulated [252]. The results suggest that delays are highly variable, contingent on whether there is line of sight communication, and the size of the transmission range (assuming that a lower range implies multipath components cannot be distinguished from each other). The Bit Error Rate (BER) is found, by simulation, to vary significantly with the relative speeds of the transmitter and receiver. High vehicle speeds mean that the channel characteristics may change in the middle of a packet transmission, which increases the BER as training sequences only exist at the start of each packet. Recently, experiments have been carried out to analyse how Doppler shift in the 5.9 GHz is related to vehicle speed [32]. This found that lower speeds and lower transmission ranges are disrupted less by the Doppler effect.

Previous experimental work has therefore mostly taken place under motorway conditions, at high speeds, and has concluded that speed does not influence 802.11b/g throughputs. The work described here differs from previous studies in that it analyses the performance of 802.11a in a semi-urban environment at low ($<$50 km/h) speeds. These velocities are particularly important as they are the norm in a congested urban environment, where there is likely to be the greatest density of access points. It is also the case that in cities users are more likely to require extra information to be downloaded to their vehicles, such as more detailed maps or traffic information, and hence will need such connectivity.

3.5.3 Experimental Set-up

Laptops were used as the transmitter and receiver for all experiments. One machine was connected using 100 Mbit/s wired Ethernet to a Cisco 1200 series wireless access point. This had an 802.11a module installed, transmitting on a frequency of 5.32 GHz. The other laptop used an NEC Warpstar 802.11a/b/g PCMCIA interface card, connected using 2 metres of low RF-loss cable to an omnidirectional aerial that provided a gain of 7 dBi for frequencies in the 5 GHz band.

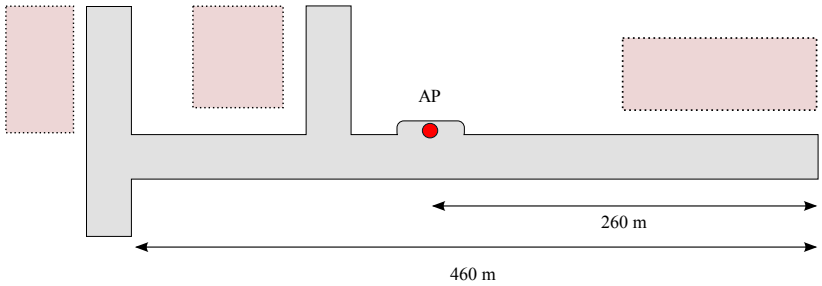


Figure 3.24: Set-up for outdoor experiments. Except for the buildings shown, the surrounding land consisted of fields. The road was two-way with low volumes of traffic.

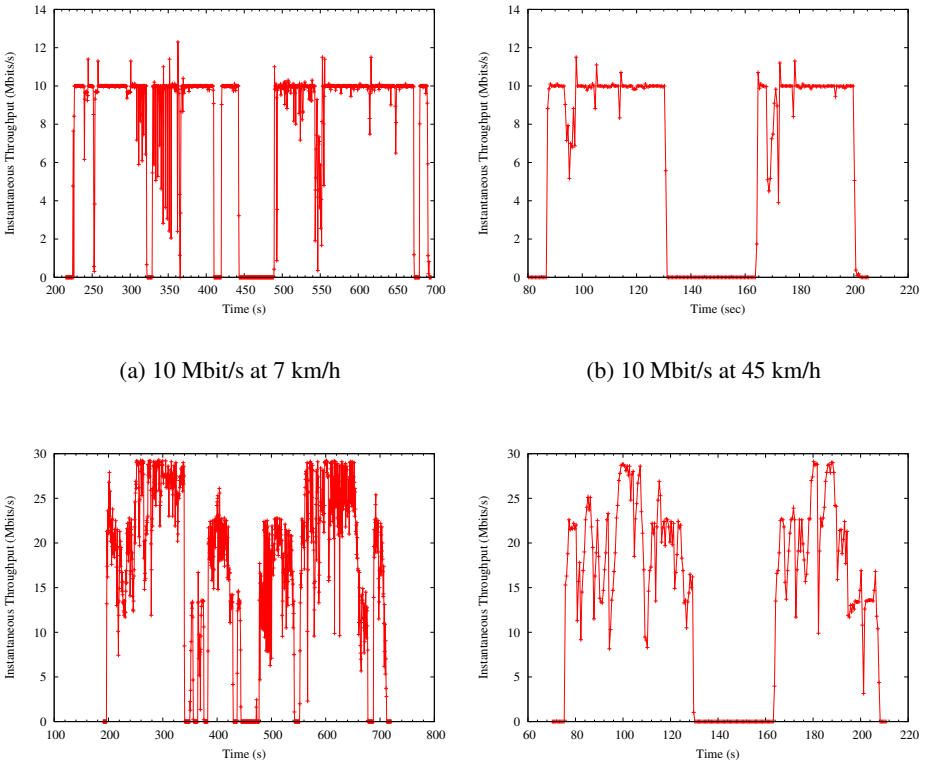
All tests were performed using the iperf[15] network measurement tool, with the server being the laptop with the 802.11a PCMCIA card, and the client connected to the access point. In each test, UDP datagrams of 1470 bytes in size were transmitted from the client to the server, and the average throughput (of uncorrupted packets received by the server) for each 0.5 second interval calculated. The two offered loads (i.e., the rate at which the client sent packets to the access point) were 10 and 30 Mbit/s.

The access point was placed on a aluminium stepladder, at a height of 1.8 metres above the road, with the laptop connected to it by 3 metres of Ethernet cable. The access point's antenna was placed with its largest face orthogonal to the direction of the road, to achieve the best possible propagation characteristics. The laptop acting as the server was placed inside the Sentient Van, and the antenna externally attached to the roofrack. The test environment was a straight, two-lane road, with low levels of traffic. There were few buildings close to the road, and these were only on one side, with the other being bounded by fields. The distance between the start and end points of each drive past the access point was approximately 460 metres, with the access point being placed in a lay-by 270 metres from one end, as shown in Figure 3.5.3. The end points were chosen such that the vehicle was out of wireless coverage when it reached the extremes of each test run.

3.5.4 Performance at Low Speeds

A total of 25 experiments were performed (the data for 4 of them being discarded due to the presence of a bus on the road) assessing the performance of a connection from a moving vehicle to the access point beside the road. Two speeds were concentrated on: 7 km/h and 45 km/h. The former is characteristic of a heavily congested inner city road and the latter of a similar road in a free flow situation. The speeds were kept as constant as possible using the vehicle's speedometer, and were verified using an onboard GPS receiver. The results from two sample runs

[15] `http://dast.nlanr.net/Projects/Iperf/`



(a) 10 Mbit/s at 7 km/h

(b) 10 Mbit/s at 45 km/h

(c) 30 Mbit/s at 7 km/h

(d) 30 Mbit/s at 45 km/h

Figure 3.25: UDP throughputs for two oppositely-directed drives past an 802.11a AP at each of various offered loads and vehicle speeds.

at an offered UDP load of 10 Mbit/s are shown[16] in Figures 3.25(a) and 3.25(b), whilst those for 30 Mbit/s are shown in Figures 3.25(c) and 3.25(d).

At 10 Mbit/s there was no significant difference in the throughput at the two speeds, with both traces showing that for the majority of the time throughput was either zero or approximately 100% of that offered[17]. In contrast, at 30 Mbit/s the connection was less "binary", i.e. the number of different throughputs exhibited was more than two, and the connection exhibited a much greater variability in throughput at 7 km/h than at 45 km/h.

[16]The traces occasionally show that the throughput exceeds 10 Mbit/s. This always occurs directly after a packet loss, and is due to some packets that were from one interval arriving in the following one.

[17]Figure 3.25(a) shows only a few isolated points that are at low throughputs, but each appears as a pair of near-vertical lines on the graph.

3.5.5 Throughput Variability

For many multimedia applications such as Voice over IP, not only is total data transmitted or received important, but also the variability of the throughput over the lifetime of the connection. Figure 3.26 shows histograms of the throughputs achieved at the two test speeds, with offered loads of 10 and 30 Mbit/s.

At 10 Mbit/s, the spread of throughputs is very similar for the two test speeds, with a peak at 9 Mbit/s. In contrast, at 30 Mbit/s, the two histograms are quite different: at 7 km/h there are no distinct peaks, with the distribution being relatively uniform above 15 Mbit/s. Meanwhile, the distribution for 30 Mbit/s offered at 45 km/h is more bell-shaped in nature, with three distinct peaks around the middle range. This indicates that at 30 Mbit/s throughput does in some way depend on the vehicle's speed.

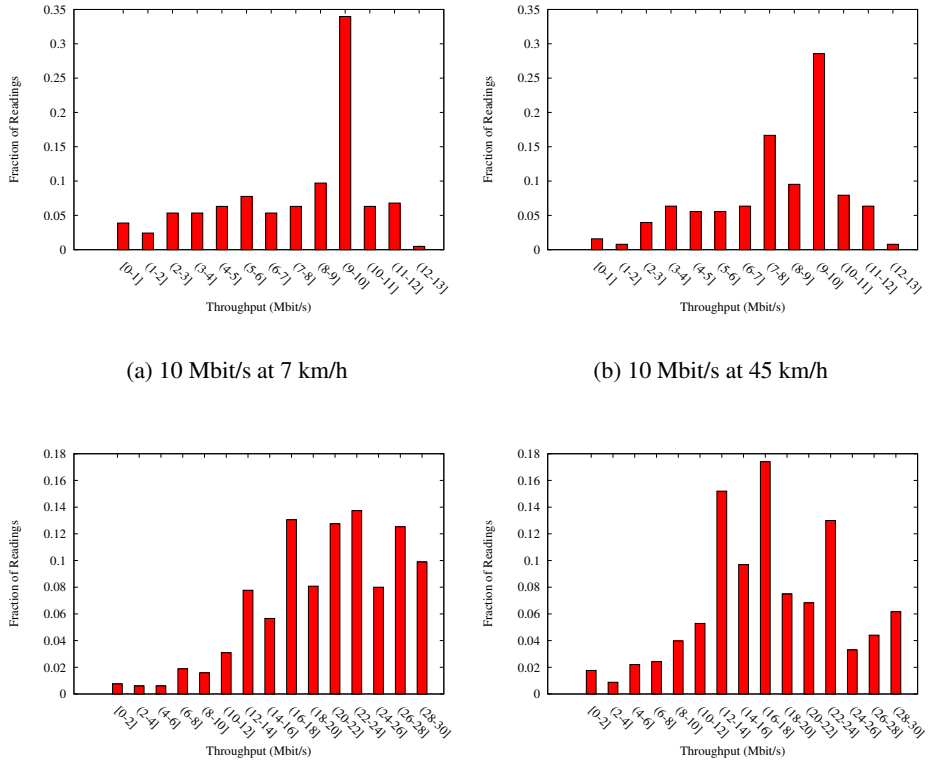
The increased variation in throughput when using 30 Mbit/s as compared to an offered load of 10 Mbit/s can be attributed to the data rate of the modulation and coding schemes in use. The transmission scheme chosen is solely dependent on the signal to noise ratio of the channel. Hence, it would be expected that the same channel data rates would be obtained at a given location, independent of vehicle speed. Whilst this is the case for an offered load of 10 Mbit/s, (and hence Figures 3.26(a) and 3.26(b) are similar to each other), it does not appear to be the case for an offered load of 30 Mbit/s.

The histograms for 30 Mbit/s shown in Figure 3.26 are evidence that travelling at *lower* speeds, whilst providing a greater connected time period, also yields a less *certain* throughput (i.e. the distribution of throughputs experienced is more uniform), but this is only evident at high offered loads. This is a significant observation, given that other related work has found that higher speeds do not affect the achievable throughput. An explanation for why this takes place is outlined below.

3.5.5.1 Defining Null Zones

In any channel that can be modelled as having Rayleigh fading (Section 2.3.1), the received signal strength will vary over distance, with regions of deep fading being present where transmitted waves interfere destructively. In environments where multipath is significant, such as those with urban street furniture, interference effects are pronounced. In contrast, in environments such as the desert or a road of very low utilisation, multipath (and hence deep fades) is less common.

Whilst intuitively it is clear that a lower received signal strength will result in a lower throughput, this is complicated by the use of modulation schemes of varying robustness. If the transmission includes sufficient error correction, a constant data rate can be maintained for large variations in RSS. Controlled, indoor experiments were therefore carried out to ascertain how IEEE 802.11a performed when the receiver antenna was placed in a known deep fade, or *null zone*.



(a) 10 Mbit/s at 7 km/h

(b) 10 Mbit/s at 45 km/h

(c) 30 Mbit/s at 7 km/h

(d) 30 Mbit/s at 45 km/h

Figure 3.26: Histograms of UDP throughputs, from all experiments at each speed and offered load, combined.

Null zones have dimensions of the order of tens of millimetres in size, and therefore require careful antenna placement. Using a spectrum analyser connected to an omnidirectional antenna, a location was identified where portions of the frequency band in use suffered from multipath interference, and hence where the overall RSS was significantly lower than its surroundings. The connection was then moved from the signal analyser to an 802.11a receiver card, maintaining the antenna in the same position. Figure 3.27 shows the variation in throughput as the antenna entered the null zone and was then removed, for an offered load of 30 Mbit/s. The first section of the graph depicts the time when the antenna was moved slightly to find the null zone once more. The second portion is the performance with the antenna stationary inside the null zone (up to 33% degradation in throughput), whilst the remainder of the trace is the performance with the antenna outside the null zone. Note that the throughput does not return to its normal rate immediately on the antenna exiting the null zone due to the receiver taking a small amount of time to adjust to the new channel characteristics. Subsequent temporary variations

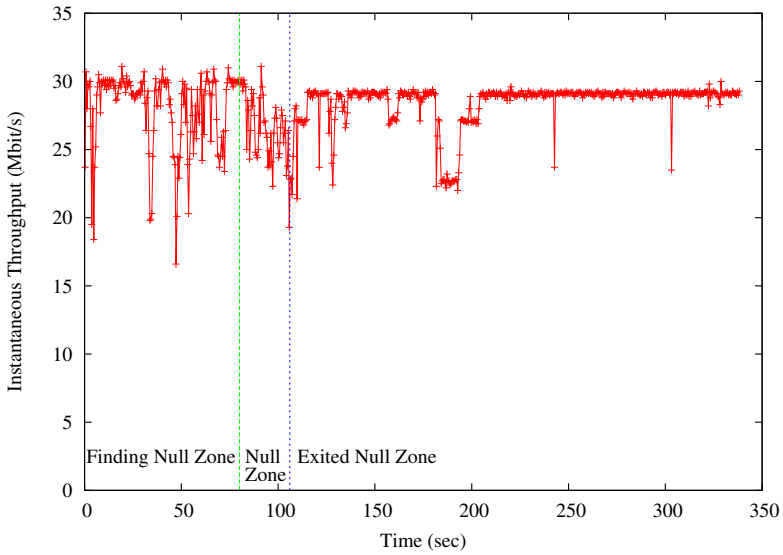


Figure 3.27: Throughput variation with the receiving antenna in a null zone, as detected using a spectrum analyser, at an offered load of 30 Mbit/s.

in the throughput are attributed to people walking along the corridor where the experiment was carried out (confirmed by other experiments).

The drop in throughput can be attributed to the transmitter selecting a lower order modulation and coding scheme when it detects a drop in signal to noise ratio. This decreases the data rate of the channel, causing the observed UDP throughput to drop. To confirm this, the above experiment was repeated at an offered load of 10 Mbit/s, where the effect of the null zone on throughput was not evident. This can be attributed to the offered load now being within the data rate of the more robust modulation/coding scheme.

3.5.5.2 Effect of Null Zones on Vehicles

In the case of a vehicle moving along a road in isolation, the number and position of null zones will only be affected by how the objects in the area are positioned, and the frequency of transmission.[18] Hence, for any vehicle speed, the range of a

[18] If other moving vehicles are present then the RF environment will be time-varying. Whilst this would make the problem more complex, the arguments presented here will still hold, as null zones will still be present.

roadside AP is constant, and the number and total length of the null zones within that coverage area is also fixed.

At low speeds, the time spent in each null zone is high enough that two effects may occur. Firstly, if the signal strength drops below a given threshold for a long enough period of time, the transmitter will select a lower order modulation/coding scheme. This will have a lower bit rate, but be more robust in order to adapt to the less favourable conditions. This selection will be reversed when signal strength is detected to have increased for a short period of time, once the null zone is exited. At higher speeds the time spent in each null zone decreases, meaning that the transmission rate is less likely to be downgraded, as the null will be treated as a transient effect, rather than a long-term drop in channel quality. Hence, the transmission rate of the channel only decreases for very short periods of time (each null zone), rather than the time required to traverse a null *plus* the time required for the rate selection algorithm to pick a higher data rate modulation/coding scheme. Such incorrect rate selection when transient packet losses or drops in SNR occur has been previously observed by others to be responsible for significant drops in throughput [26].

Secondly, at low speeds, the distance travelled by the vehicle during the transmission of each packet is comparable to the size of a null zone. For example, at a speed of 10 km/h, the time taken for a 1470 byte UDP packet to be transmitted at 30 Mbit/s is approximately 0.4 ms. During this time, the vehicle only moves 11 mm. This means that the majority of a packet's transmission could "fall" into such a region, rendering it likely to be lost. As the vehicle's speed is increased, the time its antenna spends in each null zone decreases, and hence there is a greater likelihood that the proportion of a frame affected by a null zone is small. There will come a point where the error correction in the frame will be able to correct for the interference, and hence higher layers will not see a packet drop. This explains why previous experiments at high speeds have not observed a speed-dependence in the throughputs achieved. Some authors (e.g. [210]) present graphs that do show throughputs *increasing* as speed increases from very low values, but have not provided any explanation for it. Thus, the effect presented here has been previously seen, but ignored.

3.5.5.3 Connection Times

Having examined how throughput varies with speed, another important facet of vehicular communication that must be examined is the length of time connectivity is available for. Tables 3.2 and 3.3 summarise the data from all of the experiments, only subsets of which are presented in graphical format in the figures above. In order to provide a true idea of the connection times that were experienced, the lengths of three different periods are given. Each successive period fully contains its predecessor, but also includes periods of lower throughput. The time periods were calculated using the following criteria:

Speed (km/h)	Zone 1 (s)	Zone 2 (s)	Zone 3 (s)	Runs
7 km/h	190.2 ±9.9	227.5 ±31.3	228.7 ±31.9	3
45 km/h	36.3 ±9.9	36.3 ±9.9	37.6 ±11.7	7

Table 3.2: Periods of connectivity (means and standard deviations) at an offered load of 10 Mbit/s.

Speed (km/h)	Zone 1 (s)	Zone 2 (s)	Zone 3 (s)	Runs
7 km/h	180.6 ±52.2	243.75 ±8.6	246.7 ±8.8	6
45 km/h	46.0 ±8.2	46.0 ±8.2	46.7 ±8.5	5

Table 3.3: Periods of connectivity (means and standard deviations) at an offered load of 30 Mbit/s.

- **Zone 1** of a run (the production zone) is defined as the longest time period within which the connection's throughput exceeds 10% of the offered load, with no interruptions greater than 5 seconds, excepting a single interruption of up to 10 seconds. The reasoning behind this criterion is that the majority of runs showed an interruption in their production zone, at apparently random times. It was therefore important to allow an interruption in order that meaningful conclusions might be drawn from the results. The cause of such interruptions may have been due to occasional shadowing of the access point by other vehicles.

- **Zone 2** is defined as the time period, regardless of interruption, that the connection's achieved throughput was greater than 10% of the offered load. This threshold was qualitatively chosen to span the region over which the connection is deemed to be of use.

- **Zone 3**, sporadic connectivity, is similar to zone 2, save that any achieved throughput above zero may be counted.

The data show that the length of each connection zone for a given speed is approximately constant, (the means are well within two standard deviations of each other) irrespective of the offered load. Because of the way in which zone 1 is defined, the standard deviation can be relatively high, whereas Zone 2, (for which throughput is approximately 10% of the offered load) is a more reliable indicator of performance.

Another observation is that the connection times achieved at 45 km/h were less than would be expected were they to be directly proportional to the speed of the vehicle. In other words, the connection time achieved at 7 km/h decreased in proportion to the change in speed. The explanation for this is that whilst the connected time period is proportional to the speed, this is only true once the AP has been detected and the station registered. This is dependent on the length of time taken for the MAC association frames and acknowledgments to be sent to/from the AP. In the

Antenna Position	Zone 1 (s)	Zone 2 (s)	Zone 3 (s)	Runs
Front	194.7 ±48.7	240.2 ±7	242.5 ±4	3
Back	180.5 ±65.6	247.3 ±9.9	250.8 ±11.1	3

Table 3.4: Periods of connectivity at 7 km/h, offered load of 30 Mbit/s, with different antenna positions.

course of the experiments, the re-association time (taken to be the time from which the connection is lost to when it is usable for sending UDP packets once more) was found to be 8 seconds. This then means that at higher speeds the vehicle covers a greater distance within the coverage region whilst association is taking place, leaving less time for useful communication.

3.5.5.4 Antenna Positioning

Changing the position of the antenna on the vehicle did not have any significant effect on the connection times, confirming the results reported in [92]. Two locations were used, one at the front of the vehicle nearest the passenger side, attached to the roof rack, and the other diametrically opposite, on the back roof rack. Table 3.4 shows the respective results.

3.6 Chapter Summary

This Chapter has presented an overview of the Sentient Van as a platform for making sentient transportation as outlined in Chapter 1 a reality. The communications infrastructure deployed on the vehicle has enabled experiments to be carried out concerning how UMTS and 802.11b/g are affected by meteorological and temporal factors, finding no correlation between RSS and these parameters. In addition, testing in an indoor environment allowed controlled experiments to be performed concerning AP positioning and beacon interval, finding that centred APs perform better in wave guide environments than offset ones. Finally, outdoor tests with 802.11a at low speeds showed that the throughput exhibits a greater variability as compared to higher speeds. This is significant given that much network use is likely take place on congested city streets.

All of these experiments serve to illustrate how the performance of wireless networking technologies is difficult to predict using simulation. Signal strength data collected by the Sentient Van over three years further illustrates this. Therefore, in order to provide effective connectivity to vehicles, we must use the paradigm of vehicular sensor networks described in Chapter 2 to record network characteristics. The processing of such data into coverage maps for use by proactive handover algorithms is described in the next Chapter.

4

Coverage Mapping

In the previous Chapter, the geographical variability and sensitivity of wireless network technologies to factors such as the precise location of the transmitter, as well as the effects of time of day and meteorology, were outlined. These facts serve to illustrate how users cannot take uniform, ubiquitous network service for granted. Instead, heterogeneity in both technology and administrative ownership will be the norm in the networks people use whilst on the move. This makes the process of choosing which network to connect to increasingly complex. This Chapter describes a novel method of informing such choices by proposing algorithms to process coverage-related sensor data. Coverage maps for UMTS and 802.11b/g are constructed for the city of Cambridge using data from the Sentient Vehicles project. They are then evaluated for their accuracy and compactness using further real traces collected by the Sentient Van.

4.1 Introduction

Today, many different wireless network technologies exist that could be used together to provide near-ubiquitous connectivity to vehicles. The majority (including IEEE 802.11x, UMTS cellular, and WiMax), make use of adaptive modulation and coding, and hence the rate selected is dependent on the signal strength experienced by the mobile terminal (Section 2.3.3). As a consequence, the coverage areas of the highest throughput networks are becoming ever smaller (e.g. UMTS HSPA cell coverage is less than that of GSM GPRS, whilst 802.11g has a lower range than 802.11b). There is vast diversity in the networks available: the CarTel project [24] recorded over 32,000 distinct WiFi networks in Cambridge, USA, whilst other work found some city APs whose coverage overlapped with that of up to 85 others [3]. These considerations make network selection complex.

A perennial question concerns whether a mobile user need connect to a network other than the cellular one. Given that wide-area networks cannot provide the highest throughputs ubiquitously, if a user wishes for such high throughputs, multiple local-area networks must be used. In addition, the economics of (e.g.) cellular and community WiFi networks are completely different, as are the latency and packet drop characteristics. Hence, for many applications, only using a cellular network is unlikely to be suitable.

The main problem with using multiple heterogeneous wireless networks over time is selecting when, and to what network, to perform a handover to. As described in Section 2.6.3, handover schemes may be *reactive*, where the target network is selected on instantaneous measures such as signal strength, or *proactive*, where extrinsic information concerning the networks is used. In particular, knowledge of the coverage areas of the many networks available can enable mobile clients to increase their QoS significantly. For example, awareness of the sizes of regions of radio shadow means that clients are able to decide whether to handover to another overlapping network, or whether the disruption caused by the temporary radio shadow is less than that which would be caused by the handover. How such coverage maps are constructed is the subject of this Chapter.

In this work, the focus is on the constrained problem domain of vehicles, rather than (as much of the previous work has been) on unconstrained pedestrian mobility. The rationale behind this choice is that most long distance or high speed mobility takes place on vehicles, and hence it is here that optimising handovers will be most challenging. The vast majority of vehicles move on well-defined routes, such as the road network, railway lines, or air corridors. By constraining the problem to considering such routes, this work shows how it is possible to arrive at a solution that is both compact and efficiently queried, whilst addressing a significant problem domain.

In this work, it is proposed that coverage maps should consist of an augmented directed graph representing the road network. This will ease their porting to the wide variety of personal navigation devices that are already on the market. Such devices already store a variety of metadata associated with each road, including turn restrictions, speed limits, and (in some cases) photographs or 3-D models of the surroundings. Hence, adding to the metadata a description of available wireless networks appears feasible. If navigation devices store details of the coverage of *each wireless network*, this will enable connectivity for vehicles over a desired route between two locations to be optimised. Such optimisations can be application-specific, and make use of multiple metrics such as predicted throughput, number of handovers, and data traffic charges. However, in order make such an approach feasible, a coverage map must be represented as space-efficiently as possible, to ensure that minimal extra resources are required. Storage resources are becoming less important on motor vehicles such as cars, but are still significant for handheld devices such as mobile phones or personal navigation units. Moreover, in order to achieve good performance when *querying* coverage maps (as described in detail in Chapter 5), lower complexity will generally mean fewer processing resources are required.

In this Chapter, five algorithms are described and evaluated that can be used in order to produce coverage maps from large quantities of received signal strength data. This Chapter focuses solely on the techniques used to process the raw RSS data into a coverage map, and does not go into detail about the mechanisms behind the uses of coverage maps, which is left to Chapters 5 and 6.

4.1.1 Using Coverage Metadata on Vehicles

Making *use* of metadata concerning wireless networks is an area where relatively little work has been carried out. Much of it assumes knowledge of the coverage areas of base stations in order to carry out proactive handovers.

In the Hybrid Information System [209], handovers were aided by coverage information stored in a database at each base station. This was populated over time by nodes reporting the RSS values of available networks. However, the system did not have a mechanism for amalgamating these readings, and was therefore similar to WiFi location systems, as described below in Section 4.4.5. In addition, how well the scheme would perform in a real deployment is unknown.

Meanwhile, Baig *et al.* assumed knowledge of a mechanism that predicted when connectivity would be unavailable [10]. They then simulated the effect of the error in this prediction on the performance of Freeze TCP [90]. This modified transport protocol reduced its advertised window to zero when a handover was about to occur, thus reducing packet losses. Their results suggested that good predictions were particularly important when large window sizes were used, or high outage probabilities were expected.

An approach specifically targeted at vehicles consisted of maintaining a database of locations where handovers were carried out on past journeys. This was then used to probabilistically predict handover events on future journeys [213]. The authors argued that no coverage map was necessary (only the handover database). This assumed that the locations where handovers were forced were the optimal ones: in many cases (see Section 1.3.2) a coverage-aware algorithm is likely to perform better.

Another scheme using knowledge of the road topology and traffic light timings to reserve bandwidth for expected handovers [146] assumed that base stations were aware of their coverage areas. The authors found (by simulation) that such reservation could reduce the probability that a call was dropped.

Most recently, the Mobisteer project [171] used a steerable-beam antenna to record, for each road segment, the wireless network with the lowest packet drop rate. On subsequent journeys, this network was automatically connected to once more. Whilst this scheme achieved better mean physical layer throughputs, it neglected the impact of the large number of handovers it recommended. Also, without knowledge of the coverage areas of each network, it again assumed that a reactive scheme was best.

The work described in this Chapter differs in that using a coverage map it is possible to calculate the *optimal* location at which a handover should be carried out. Moreover, it allows the cost of a handover to be compared to the gain in connecting to the target network (which can be calculated once that network's coverage is known). Hence, better handover decisions can be taken. For example, a reactive

algorithm will choose the best network (according to some metric such as RSS) perhaps once every second. This may mean that a brief three second region of radio shadow in the coverage of one network results in a handover to another. Such a handover might involve a disconnection of five seconds (and hence cause more disruption than the region of radio shadow would have done). However, it *will* take place because the reactive algorithm has no knowledge of the extent of the radio shadow, and judges the "best" network to be that with the highest RSS.

4.2 Data Collection & Hardware Specificity

As described in Chapter 3, a large quantity of sensor data was collected by the Sentient Vehicles project, where a vehicle was driven by many members of the author's research group for their day-to-day activities around the city of Cambridge. In particular, data concerning the signal strengths of UMTS cellular networks and IEEE 802.11b/g local area networks was collected, along with associated GPS location information.

Both the UMTS and 802.11b/g cards in the vehicle yielded hardware-specific RSS measurements, i.e. these would be different for a different hardware configuration. This does not constitute a problem, as were another set of equipment to be used, the sensitivity of the new configuration could easily be determined and compared to that originally used. Previous work by Haeberlen *et al.* showed that the relationship between the RSS values reported by different 802.11b/g cards is linear and simple to determine [99], whilst the 3GPP TS 127.007 standard [74] provides a conversion from the unitless UMTS RSS values reported by the cellular modem into signal powers in standard units of dBm. RSS readings from different HSDPA modems should therefore be approximately consistent. Hence, a coverage map based on readings from one hardware configuration could be trivially adjusted to suit an another.

In addition, by collecting RSS data, rather than instantaneous throughputs achieved, the utility of the coverage maps was not limited to a particular technology. For example, if a high RSS were available at a particular location, 802.11b hardware would achieve a raw throughput of 11 Mbit/s. For the same RSS, 802.11g would achieve up to 54 Mbit/s. Thus, coverage maps are more generally applicable when they contain RSS values.

Finally, a throughput (rather than RSS) map would be specific to a particular protocol (in particular TCP or UDP), packet size, and forward error correction rate, and hence would not be as generally applicable. In contrast, RSS data are only subject to the physical effects on the radio channel, such as attenuation and interference[1], which will be present no matter which higher-layer protocol utilises the channel.

[1] It should be noted that in environments where there is significant multipath, certain portions of a wideband signal may be degraded, which may not result in a large RSS decrease but still impair throughput. This effect *would*be visible were throughputs to be recorded instead.

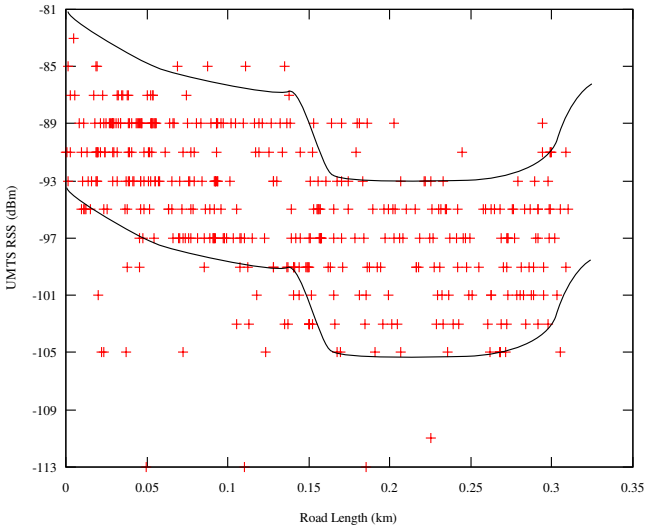


Figure 4.1: Input UMTS RSS data (452 points) for an exemplar road. The envelope formed by the two (identical) lines on the graph encloses 90% of the data points, and has a constant height of 12 dBm.

4.3 Signal Strength Variability

Coverage maps implicitly assume that RSS readings are stable (or vary deterministically) over time. In Section 3.2 it was shown that the RSS measured at a given location is subject to random noise. However, there is no correlation between the variation observed and time of day or various meteorological factors. Other research such as Intel's Place Lab project has also reached similar conclusions concerning the stability of RSS readings for a given location for IEEE 802.11b/g [33, 155].

The results in Section 3.2 showed that both UMTS and 802.11b/g RSS values for a given location can be approximated by normal distributions, having standard deviations of 3 dBm and 3.5 dBm respectively (hence 90% of values will be within 6 dBm and 7 dBm of the mean, respectively). Therefore, in this Chapter it will be assumed that at a given location RSS has a single "true" value, which is perturbed by noise taken from a distribution with zero mean and with the relevant of the above standard deviations.

An example of the input data collected by the Sentient Van is shown in Figure 4.1. The trace shows the UMTS RSS values collected on multiple journeys over a single road. The two identical lines drawn on the graph define an envelope which has a constant height of 12 dBm, containing 90% of the data. This is as expected, given the model of the distribution of UMTS RSS proposed above. This degree of natural

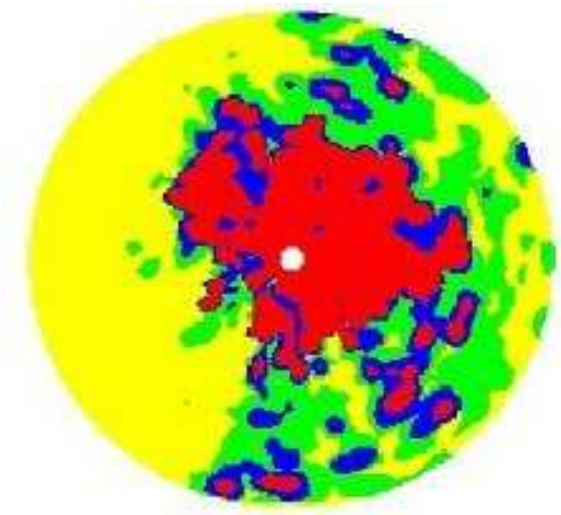


Figure 4.2: The coverage area of a particular omnidirectional antenna. Low RSS values are indicated by light colours. Diagram used with permission from Comgate Engineering Ltd., Canada (August 1999).

variation is important for putting into perspective the error in the predictions made by coverage maps, and will be used in the evaluation in Section 4.8.

4.4 Existing Coverage Mapping Methods

Wireless access point mapping has long been carried out as a development of "war-driving", with web sites devoted to recording where users can obtain free network access. Mapping is also useful for network providers when ascertaining where new equipment needs to be deployed, or analysing spectrum use [25]. Maps record the location of infrastructure, and then make general assumptions about the coverage areas, e.g. a disc-shaped zone of radius 100 metres for a WiFi hotspot. These assumptions are frequently found to be untrue to real life [137], as shown in Figure 4.2. However, this is not a significant problem for for pedestrian users, who move until they find coverage from a hotspot, and then remain stationary.

A variety of techniques have been suggested for generating coverage maps that are more detailed. In most cases, the RSS and/or throughput is surveyed at a number of locations, and these are then used to predict the coverage at nearby locations which were not included in the survey. A selection is overviewed below.

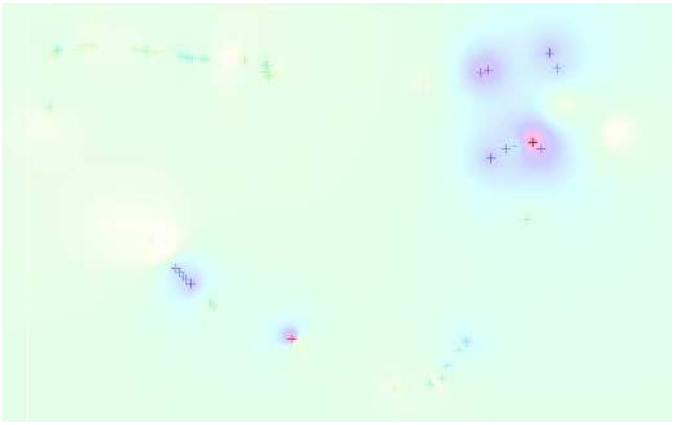


Figure 4.3: Inverse Distance Weighting of UMTS RSS readings from the Sentient Van, courtesy of J. Davies. Darker colours imply higher values of RSS.

4.4.1 Inverse Distance Weighting

Inverse Distance Weighting of points assumes that the RSS (or other data type) value at any point in space is related to the values at other points in space solely by the distance they are apart. Hence, by taking a weighted sum for each point, where higher weightings are assigned to sample points close to the point we wish to predict the value of, and lower weightings to samples further away, a coverage map is obtained. An example that makes use of a small number of UMTS RSS readings collected by the Sentient Van is shown in Figure 4.3, and further mathematical details can be found in Section 4.6.1. Several problems arise with this method:

- **Topology is not taken into account.** The terrain or building profiles of the area are ignored, i.e. it is assumed that the only impediment to signal propagation is distance. Hence, regions of radio shadow that may be very close to a transmitter may not be seen.

- **Time variability is ignored.** Generally, work using linear interpolation relies on making a relatively large number of geographically distinct measurements, but does not repeat those measurements at the same locations. Such a technique inherently assumes that the RSS at each location is single-valued.

- **Storage/Querying is problematic.** Whilst storing the human-readable image generated by an interpolation procedure is simple, it is space inefficient. Also, answering a query requires locating the point in the database that is geographically closest, which is inefficient for large data sets.

Kamakaris and Nickerson [127] published human-readable contour maps of a university campus generated by linear interpolation, surmising that such data could be converted to graph form by overlaying the road network on the contour map (see below) and assigning a single throughput value to each road. However, they did not go on to implement or evaluate this method.

4.4.2 Contour Simplification

Generating contours over a set of sample points is relatively simple, and involves locating all points of equal RSS and calculating their convex hull, as shown in Figure 4.4. Sometimes, a pre-processing step is applied where Inverse Distance Weighting is applied to generate a grid of evenly spaced points, from which contours are then generated. However, such approaches also rely on the RSS value at a single point not varying (as evidently the contour calculated would vary depending on the values supplied). In order to aid storage, contour polygons are simplified to decrease the number of points that are needed to represent the contour. Lück *et al.* carried out contour map generation and contour simplification [152] by simulation rather than measurement: a wireless access point was considered to have a boundary within which it provided service, and outside which it was unusable. The simplified contours were then stored in a database that allowed clients to query what the service boundaries of each access point were [153]. Unfortunately, no results have been presented concerning how accurate the predictions made by the system are.

4.4.3 Kriging

Another more complex interpolation technique is kriging [139], an approach originating in the field of geology. Here, a variogram is constructed, which is a function that expresses the spatial interdependence of two points (i.e. how the value of one depends on the value of the other, and on the distance they are separated by). This can then be used to extrapolate from a set of input points, such as the locations of cellular base stations, to a coverage map of a large area. This approach places great emphasis on the correctness of the variogram, which appears to be difficult to construct for a complex urban environment. The approach described in [56] compared over 6,000 measured values with those generated by Kriging, and examined where the errors were largest. Statistics of the prediction errors were not given, and hence it is difficult to gauge the effectiveness of the approach.

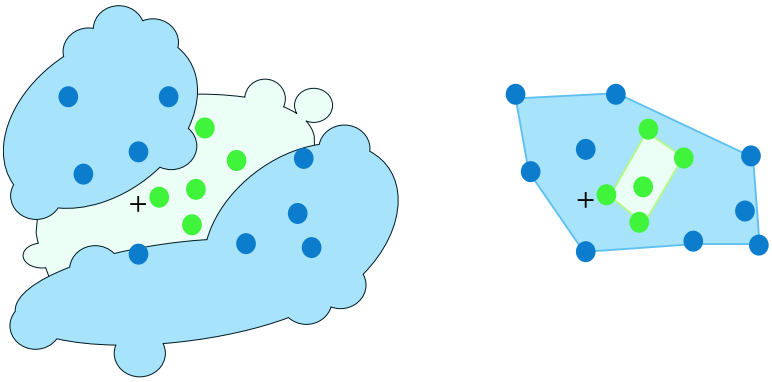


Figure 4.4: Representing coverage maps using contours. The left-hand diagram depicts the actual coverage of a base station, where colours represent the RSS value, together with points where measurements have been taken. The right-hand diagram depicts how the contours that are inferred from the measurements can give an incorrect notion of the coverage available. The location marked by a cross is in an area of light colour coverage. However, the simplified contour representation places it within the area of dark colour.

4.4.4 Propagation Simulations

Departing from large-scale interpolation techniques, there are a number of approaches that have used grids of cells, each with its own associated RSS value, as coverage maps. These cells may be of fixed or variable size, and their values are updated whenever a new reading is obtained, as shown in Figure 4.5. Moreover, radio propagation path loss models can be used in order to estimate from a reading in one cell what the corresponding values would be in neighbouring cells [145]. However, these approaches assume that all necessary geographical topology information is available to be provided to the path loss models. This is unlikely in cities, where wireless propagation is complex (particularly due to building heights [79]), unless each grid cell is of the order of a few square metres in area.

4.4.5 Relation to Wireless Positioning Systems

As described in Section 2.7, wireless positioning systems are another domain for which large quantities of RSS data is collected. The principal difference between the work described in this dissertation and research into wireless positioning systems is that the objective of this dissertation is to create highly *space-efficient* rep-

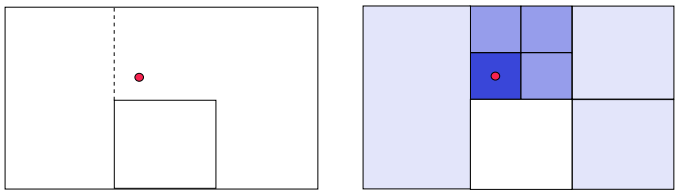


Figure 4.5: Using the Neighbouring Cells technique from [145]. The red circle represents a wireless AP. The left-hand diagram shows the building layout, with solid lines representing thick concrete, and the dashed line glass (through which the signal propagates). The right-hand diagram shows the RSS (darker implies a higher RSS value) and how this is represented using cells of different sizes.

resentations of the RSS data.[2] This work also seeks to make this space-efficient representation in such a way that it is useful for proactive handover algorithms, i.e. to be able to efficiently answer questions such as "what is the coverage area of network x?", or "is it more disruptive to perform a handover to network y due to a known region of radio shadow on network x?". Clearly, the data sets used in Place Lab [143] (and similar systems) could be processed to answer such queries: the crucial question addressed in this Chapter is *how such processing is done*.

4.5 Problem Specification

The input values to a coverage mapping algorithm are in general not equally spaced, and are subject to random noise. Therefore, an algorithm should cope with such data, whilst producing maps that allow RSS to be predicted with low error, and that are space-efficient.

Linked to the second requirement is the idea that the raw sensor data will be uploaded to a central authority (or an authority responsible for a particular area), and processed by that entity using the algorithms described below. The resulting coverage maps are then distributed back to the vehicles. Hence, vehicles are *not* required to have large quantities of computing resources, as to utilise a coverage map should have a similar overhead to using the maps in a traditional satellite navigation unit.

In contrast to other work on coverage mapping, this Chapter focuses specifically on mapping RSS on roads, rather than all space. This constraint allows these maps to be efficiently stored by representing the coverage along each road as a line, rather

[2] The Place Lab wireless location system project has already noted [143] that some devices may be too resource-constrained to download the entire fingerprint database for the city they are located in.

than a surface.[3] By reducing the problem to one of line simplification, coverage *extents* can be produced. These consist of a tuple $(v_{\text{start}}, v_{\text{end}}, l_{\text{start}}, l_{\text{end}}, t)$ composed of a start RSS value (v_{start}), an end value (v_{end}), a start co-ordinate (l_{start}), an end co-ordinate (l_{end}, measured as the proportion along the total length of the road), and the timestamp of the most recent data point used to create the extent (t). An extent signifies that between the start and end points the value of the sensor concerned (RSS in the case of coverage maps) varies linearly from the start value to the end value. Hence, the output of any smoothing process should be a small number of contiguous extents spanning the road's length.

4.6 Novel Methods of Coverage Mapping

In order to satisfy the above criteria, four algorithms were implemented which had not been traditionally applied to this problem domain. One of the contributions of this dissertation is the adaptation of these algorithms from their original uses in fields as diverse as chemical spectroscopy and 2-D shape simplification to the construction of coverage maps. Each algorithm is briefly described in turn, followed by the author's adaptations to it. The performance of each one is then evaluated in Section 4.8, and compared to an established algorithm in the field.

4.6.1 Nearest Neighbour Interpolation

The simplest (but most processing intensive) technique for constructing extents is to pick sample points along the road in question, estimate the value of the quantity under investigation at the sample point, and then generate extents from those sample points. This was carried out using nearest neighbour inverse-distance weighted interpolation, as proposed by Shepard [206]. This is one of the few interpolation algorithms that is able to utilise irregularly spaced input data, and does not snap it to a regular grid prior to interpolation. The technique has been used before for coverage mapping [127], and hence is included here for comparison purposes. The adaptations made to it were only to make it suitable for generating extents.

4.6.1.1 Original Algorithm

Sample points are selected that are separated by a particular interval, λ, along the road's length, such as every 100 metres. Regardless of the length of the road, a sample point is picked at $l = 0$ and another at $l = \text{roadLength}$. For each sample point the database is queried to find the set of data points, S, that are within a

[3] Note that this is only valid for roads that are not very wide. The majority of roads in Cambridge, UK are a single lane in each direction. For larger multi-lane situations the algorithm could be performed on a per lane or per carriageway basis.

certain radius α (set to 10 metres), above which they are considered too far away to be correlated with this sample point. The value, v_j, of the jth sample point (ordered by length along the road) is then calculated as:

$$v_j = \begin{cases} \frac{\sum_E s_i}{|E|} & \text{if } E = \{s_i \in S | d_i \leq \epsilon\} \text{ and } |E| \neq 0 \\ \sum_{i=0}^{|S|} \frac{s_i}{d_i^2} & \text{where } \alpha \geq d_i > \epsilon \text{ otherwise} \end{cases}$$

where d_i is the distance from data point i to the sample point under consideration, and s_i is the value at data point i. The first condition assumes that all data points at a distance less than or equal to ϵ (set to 1 metre) are considered to be at the location of the sample point, and are hence averaged in preference to weighting the values of nearby neighbours. It should be noted that this has the possible drawback that the value of a sample point could be set to that of a nearby data point that was an outlier. However, it can be argued that this data point is at (or is very near to) the sample point, and hence should be regarded as the authoritative value.

4.6.1.2 Adaptations

Having calculated values using nearest neighbour interpolation for all the sample points along a road, they are then amalgamated into extents on the basis of how different their values are. Initially, the first extent represents only the first sample point, v_0. To amalgamate further sample points into it, the mean of the sample points currently represented by the extent, $\overline{v_{0..j}}$ (where in this case $j = 0$) is taken, and compared to the next sample point to be amalgamated, v_{j+1} (in this case v_1). If $\overline{v_{0..j}}\gamma \geq |v_{j+1} - \overline{v_{0..j}}|$, where γ is in the range [0,1], then v_{j+1} is amalgamated into the current extent. The higher the value of γ the greater the allowed difference between the current mean value of the extent and the next sample point that may be averaged together. In this implementation γ is set to 0.2. The resulting extent's start point is $l_{\text{start}} = \max(l_j - \frac{\lambda}{2}, 0)$, where l_j is the position of the jth sample point as a fraction of the road length. Its end point is $l_{\text{end}} = \min(l_{j+1} + \frac{\lambda}{2}, \text{roadLength})$. The amalgamation process continues until there are no further sample points or γ is exceeded, in which case a new extent is begun, and amalgamation restarts from that sample point. A special case occurs if $\text{roadLength} < \lambda$, in which case the value of a sample point at the mid-point of the road is obtained, and then the mean of the start, end and mid-point sample values is deemed to be the value for an extent spanning the entire road.

The resulting extents are pairs of start and end lengths along the road, with a single associated sensor value. Hence, when stored in the database, the extent is $(v, v, l_{\text{start}}, l_{\text{end}}, t)$.

There are two problems with this approach; firstly, there may be a large number of points within distance α of a sample point that make the above process very time consuming. Secondly, picking sample points at a regular distance λ risks

smoothing out significant features between the sample points. Whilst λ could be dynamically varied, this would require a knowledge of the surface to be sampled, which is in essence what is being attempted with this algorithm.

4.6.2 Dominant Point Detection

Once the RSS readings have been snapped to lengths along a road, the problem metamorphoses from three dimensions to two, as we must now deal with how the RSS *profile* of the road may be processed. Were all the points in the profile to be joined together, the result would be a line graph in the 2-D plane formed by the length along the road axis, and the RSS axis. The task is to simplify the line graph to obtain a compact, yet accurate, representation.

As described in Chapter 3, the RSS readings obtained at a particular location can be thought of as being derived from a single value perturbed by random noise. Hence, when representing the RSS profile of a road, the goal is to find an approximation to the curve that best fits the noisy data. Unfortunately, the best fit curve may well not be C_1 continuous, i.e. its gradient may not vary smoothly in some cases. For example, a radio shadow caused by a building might appear as a step change on the graph. This lack of smoothness is a characteristic that renders many traditional approaches inapplicable, as detailed in Appendix A. As a consequence, the concept of *corner detection* was applied to the problem.

4.6.2.1 Corner Detection

The idea of corner detection can be applied to the detection of the most important features of the RSS data. Corner detection was originally developed in order to derive simplified representations of two-dimensional *closed* curves (i.e. polygons), with an example shown in Figure 4.6. An early use of these algorithms was in detecting where in an image the arms of chromosomes were located, in order to perform automated measurements of their size [83].

Initial work in 1970 concerned the use of n-chains [83], where each point was assigned an integer based on the angle of the line between the points preceding the point in question, the point itself, and those succeeding it. Rosenfeld and Johnston [194] then proposed an algorithm that detected points which had significant curvature by calculating the cosine of the angle at point p_i made by the lines $\overline{p_{i-k}p_i}$ and $\overline{p_ip_{i+k}}$, known as the k-cosine curvature. This work was later developed to average the cosine curvature of each point with those of its neighbours, which yielded better results [195]. These methods required the value of k to be set correctly, depending on the scale of the smallest feature in the curve. An incorrect value of k would mean that the algorithm's choices of corner points would be poor.

This requirement for knowledge of the scale at which features were present rendered the above algorithms inapplicable to the general case. A variety of proposals

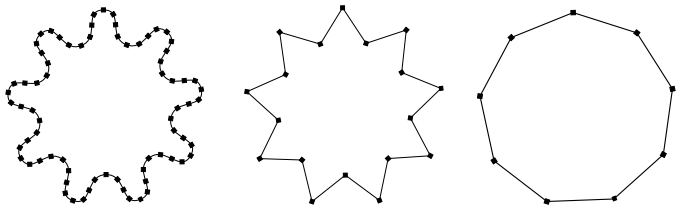


Figure 4.6: Dominant point detection. The left-hand diagram shows the original closed curve to be simplified. The middle shape is a representation using 18 points, whilst the right-hand one shows a lower fidelity, but higher compression ratio, 9 point approximation.

were made to achieve scale-independence: Lowe [151] proposed a method that yielded approximation curves at multiple scales, the best-fitting of which was selected as the final output. Teh and Chin published an algorithm that required no input parameters, but instead took advantage of the advances in processing resources to examine curvatures at many more values of k than could have been done previously [220], finding the curve's *dominant* points. Wu developed this further by making the value of k at each point depend on what its value had been for its preceding neighbour [248], surmising that the frequency of variation in a curve was likely to be similar at closely located points. Meanwhile, Marji *et al.* adjusted Rosenfeld and Johnston's algorithm in order to make the results less dependent (but not independent) of the value of k supplied [160]. For this work, the Teh/Chin/Wu approach was chosen, as it results in a single output curve, whilst being scale independent.

In terms of which methods perform best, the difficulty is deciding *how* to compare them. Kadonaga and Abe examined how the choices of corner point made by 11 different corner detection algorithms varied as the input curves were transformed by scaling, rotation, and reflection [125]. Algorithms whose choices were invariant over the different transformations were assessed to be best, with the n-chains method performing well. They also assessed how the set of corner points picked by each algorithm compared to those picked by human test subjects, finding that the Rosenfeld-Weszka method performed best. In contrast, Rosin evaluated how well the points that were chosen represented the input curve [196]. Specifically, the two metrics used were its fidelity and its compression ratio. Fidelity was measured in terms of integral square error of the approximation relative to the input shape, whilst the compression ratio measured the number of points that the approximation used, relative to the original shape. Clearly, minimising both of these quantities is of benefit. Rosin evaluated 23 algorithms, finding that Lowe's algorithm performed well in terms of his metric of the product of the fidelity and efficiency of the algorithm.

It was considered that applying a technique based on dominant point detection to the RSS data might yield a useful representation, whilst simultaneously smoothing the data. Specifically, using the ability of dominant point detection algorithms to calculate the *region of support* of each point, it is possible to pick those points that have few neighbours showing the same upward or downward trend as candidates for discard. The cosine curvature of each point can also be used to detect those points that are significant departures from the general trend, and can be assumed to be noise. This smoothing element allows many candidate dominant points to be discarded, leaving the "real" ones that best represent the RSS curve. These techniques are described in depth in the sections that follow.

4.6.2.2 Original Algorithm

We define an open digital curve S as an ordered sequence of points $S = \{p_1, \cdots, p_n\}$ where each $p_i = (x_i, y_i)$, and the x_is are monotonically increasing. In the case of a closed curve, p_1 is a neighbour of p_n, as the start and end point of the curve must be identical (and hence the x_is are not monotonically increasing). When analysing the curve to find its "corners", the aim is to find the local curvature maxima, i.e. those points at which the rate of change of gradient with length is greatest. These are known as the *dominant points*.

The first stage of this process is to calculate the Freeman chain codes [80] for all the points, and eliminate those that are collinear. This is done by calculating the vectors between each pair of points, $\mathbf{w}_i = \overline{p_i p_{i+1}}$, then mapping the direction of each $\mathbf{w}_i$ onto the closest one of 8 possible directions (each separated by $\frac{2\pi}{8}$ radians) numbered from 0 to 7. The chain code for $\mathbf{w}_i$ is denoted f_i. This enables points that lie on a straight line to be easily eliminated from consideration as candidate dominant points, by discarding p_i if $f_{i-1} = f_i$.

Following linear point removal, the *region of support*, k_i for each p_i is calculated. The larger the region of support, the greater the number of input points that support the hypothesis that p_i is a dominant point. Various methods for calculating the region of support have been proposed. Here, the iterative method given by Teh and Chin [220] is presented first, followed by a description of how this is adapted by Wu [248].

We define:

$l_{i,k} = |\overline{p_{i-k} p_{i+k}}|$ the length of a chord between two points
$d_{i,k}$ the perpendicular distance of p_i from $\overline{p_{i-k} p_{i+k}}$

We initially set $k = 1$, and increase it until the condition

$$\begin{cases} \frac{d_{i,k}}{l_{i,k}} \geq \frac{d_{i,k+1}}{l_{i,k+1}} & \text{if } d_{i,k} > 0 \\ \frac{d_{i,k}}{l_{i,k}} \leq \frac{d_{i,k+1}}{l_{i,k+1}} & \text{if } d_{i,k} < 0 \\ \text{false} & \text{if } d_{i,k} = 0 \end{cases}$$

yields true[4]. The final value of k is stored in k_i, and indicates that the region of support for p_i is the set of points

$$D_i = \{p_{i-k}, \cdots, p_i, \cdots, p_{i+k}\}$$

We now calculate the k-cosine curvature [194], c_i, the cosine of the angle that the curve turns through as we traverse each D_i:

$$c_i = \frac{\mathbf{a}_{i,k} \cdot \mathbf{b}_{i,k}}{|\mathbf{a}_{i,k}||\mathbf{b}_{i,k}|}$$

Where

$$\mathbf{a}_{i,k} = \overline{p_i p_{i+k}}$$
$$\mathbf{b}_{i,k} = \overline{p_i p_{i-k}}$$

This implies that c_i will be nearer to 1 if the angle turned through by D_i is small, and tend to -1 as the angle approaches π radians.

Three further elimination steps are then performed:

- **k-Cosine discard threshold:** For each p_i, if $c_i > \mu$, eliminate p_i from consideration. This eliminates points at the centers of very broad angles, which are unlikely to be dominant points (Wu's algorithm, step 4).

- **Suppress small regions of support that are overlapped by neighbours:** For each p_i, if $k_i < k_{i+1}$ or $k_i < k_{i-1}$, eliminate p_i from consideration (Wu's algorithm, step 5).

- **Discard large angled points if adjacent to a small angled point:** For each p_i that has not yet been eliminated, if $k_i = 1$ and

  $$p_{i+1} \text{ has not been eliminated } \wedge c_i \leq c_{i+1} \vee$$
  $$p_{i-1} \text{ has not been eliminated } \wedge c_i \leq c_{i-1}$$

  then eliminate p_i from consideration (Teh-Chin algorithm, step 3c).

As a development to the Teh-Chin method of calculating the region of support, Wu proposed a dynamic method for determining the value of k [248] that involved assuming that k_i was close in magnitude to k_{i-1}. In this method, initially $k = k_{i-1}$, and on the jth iteration a value of k is tried that is j more than k_{i-1}, and another that is j less than k_{i-1}. This method for generating coverage maps was also evaluated.

[4]Note that Teh and Chin [220] give an alternative condition that may be satisfied instead. This is not relevant to this work because the values of x_i are monotonically increasing.

4.6.2.3 Adaptations

As noted above, the original dominant point detection algorithms were only intended for the simplification of closed curves (polygons). In order to apply them to non-closed curves such as graphs of RSS, two approaches were trialled. Initially the algorithm's iteration through different values of k was constrained in order that $i + k \leq n$ and $i - k \geq 0$. This meant that at the beginning and end of the curve incorrect decisions were made over whether points should be discarded. To correct this the curve was reflected in the y-axis at both ends[5], such that if the value of k exceeded the first bound given above, $d_{i,k}$ was calculated between p_i and the chord $\overline{p_{i-k}p_{n-(i+k-n)}}$, and similarly for the case of $i - k < 0$.

A further observation that was made was that the Teh-Chin algorithm performs better on sparse data (i.e. fewer points per metre of road) than does Wu's algorithm, and vice-versa for dense data. A dynamic algorithm was therefore implemented that segmented input data into regions of high and low density, and applied the Teh-Chin and Wu algorithms to the relevant sections. This is referred to below as the *Density-Dependent* algorithm. The threshold between high and low densities was empirically determined to be an inter-point separation of 17.5 cm (i.e. approximately five points per metre of road).

When using Wu's algorithm with dense data, a greater degree of smoothing was needed. Consequently, a second k-cosine discard threshold was added, μ_2. Hence, the first step after calculating cosine curvatures is modified to eliminate p_i if $c_i > \mu$ or $c_i < \mu_2$.

Taking advantage of the knowledge of the region of support of each point, the output of the dominant point detection algorithms was smoothed by discarding any points with a $k_i < \kappa$, where $\kappa > 1$ and is chosen by experimentation. This is because points with few others "supporting" them have less raw data to support the hypothesis that this corner in the curve is due to real data rather than a few outliers. However, with sparse data, regions of support will evidently be smaller (due to fewer points per unit length of road) than those of high density. Hence, in the proposed dynamic algorithm, κ is also varied depending on the density of the input data, using $\kappa = 2$ for high densities, and $\kappa = 0$ for low densities.

Finally, the output is further smoothed by removing extents that are very short. For all remaining points p_i the length of $\overline{p_i p_{i+1}}$ is compared to a threshold ζ. If it is smaller than ζ then p_i is removed from consideration. The set of candidate points is iterated over until there are no extents below the threshold. This ensures that extents that concern very small distances are ignored. Hence, ζ must be set to reflect the minimum distance over which a vehicle travelling at a plausible speed would have time to adapt its network connections in order to take advantage/cope

[5] Reflection in the y-axis means that if there exists a point at (x, y) co-ordinates (a, b), when it is reflected in the left-hand y-axis (at $x = 0$) a point is created at $(-a, b)$. Similarly, when reflecting in the right-hand y-axis $(x = n)$ we obtain $([n - a] + n, b)$.

with the change in network performance. ζ was empirically determined to be 10 metres for high densities, and 9 metres for low densities.

4.6.3 Savitzky-Golay Smoothing

Although the dominant point detection algorithms described in the previous section work well on their own, for large quantities of noisy data they are still prone to outputting either a large number of dominant points, or, if κ and μ are too high or too low respectively, too few to be representative of the inputs. Therefore, the use of a filtering step prior to executing a dynamic dominant points algorithm was investigated. This reduces the rôle of the dominant points algorithm to representing only the general shape of the curve, rather than smoothing it.

4.6.3.1 Original Algorithm

Savitzky-Golay smoothing [198], also commonly known as the least-squares or Digital Smoothing Polynomial method, is a windowed low-pass filter originally used for analysing chemical spectroscopy data. For each input point p_i a high-order polynomial is fitted to the data within the window centred on p_i using the least squares method. The value corresponding to p_i that is output is the value of the fitted polynomial at the x co-ordinate of p_i. The window is then moved to p_{i+1}, and an entirely separate least squares procedure is executed. The use of a polynomial fit contrasts with other window-based smoothing filters, where a constant value is assigned to p_i based on the average of the points within the window. Such a constant value results in local maxima and minima having their y values reduced, whilst the polynomial fit preserves these far better [189].

The algorithm as given by Press *et al.* [189] was therefore implemented, which involves straightforward matrix operations. The algorithm assumes equally spaced data, which data from vehicles does not tend to be. However, with dense data, this constraint can be relaxed to one where provided that the change in the y value of the input data over the window length is small (i.e. the majority of the points have similar values), the algorithm can be used. Hence, the algorithm is only of use where there is dense input data. In addition, a window size of 101 points is used (empirically determined to be large enough to achieve the necessary degree of smoothing). Therefore, the input must have at least this many data points in order for the algorithm to be run over it, thus excluding roads that have not been driven along multiple times.

4.6.3.2 Adaptations

The number of output points of the Savitzky-Golay smoothing step is equal to the number of input points, but the output graph now has a much smoother profile. To reduce the number of points, the data is passed into the dynamic version of the dominant point detection algorithm described in Section 4.6.2.2 to obtain a more space-efficient representation. Hence, whilst the implementation of the Savitzky-Golay algorithm is not innovative, coupling it with the dominant point detection algorithm is, to the best of the author's knowledge, a technique that has not been previously used.

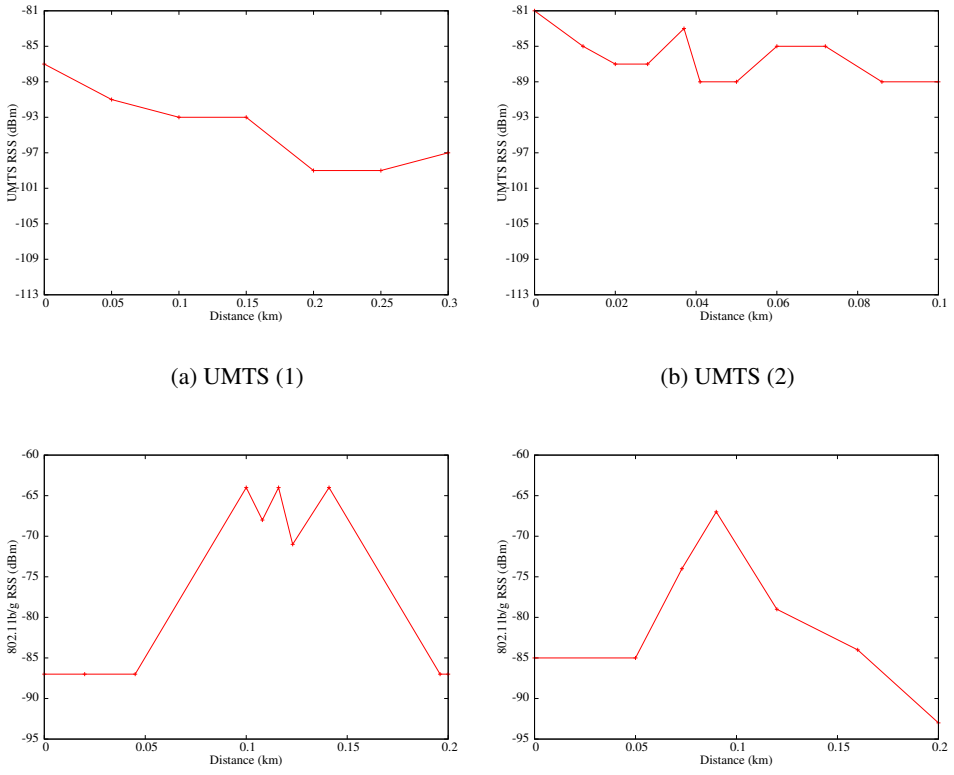
4.7 Simulation Results

Apart from the nearest neighbour interpolation algorithm, none the algorithms proposed in this Chapter have been previously used for processing RSS data. Therefore, synthetic data were generated, processed by the algorithms, and the results compared to the known values used to generate the synthetic data. This proof of concept stage also allowed the algorithms' parameters to be optimised, in preparation for their use on real data as described in Section 4.8.

4.7.1 Synthetic Data

To generate synthetic data, real traces recorded by the Sentient Van were examined, and used to create two traces each for UMTS and 802.11b/g that provided a single value for any length along the road. The data used for the lines mimicked the pattern qualitatively inferred from data for real roads as recorded by the vehicle. Each of these *source curves* is shown in Figure 4.7. The small features included in the source curves are representative of real life, and are useful for evaluating how the proposed algorithms perform with both long-distance and short-distance changes in RSS.

Points were generated from the source curves by picking locations along the length of the (synthetic) road at random. The set of such locations is termed X. The number of points in X was varied to simulate different densities of source data. The value of the source curve (denoted s) at each of these locations was calculated, i.e. $s(x), x \in X$, and then perturbed by adding noise, n_x sampled from a Normal distribution with zero mean and a standard deviation of the appropriate value (3 dBm for UMTS, 3.5 dBm for 802.11b/g, see Section 3.2), giving $p(x) = s(x) + n_x$ as the perturbed value. In this way, synthetic curves were produced that were similar to those curves seen on real test drives, but for which the true values were known. For each of the four source curves, 40 different synthetic data sets were generated, 10 at each of four different point densities (102, 250, 500 and 1000 points per 100



(a) UMTS (1)

(b) UMTS (2)

(c) 802.11b/g (1)

(d) 802.11b/g (2)

Figure 4.7: Source curves for generating synthetic data.

metres), and each having its own unique set of (random) perturbations to the source curve.

4.7.2 Evaluation Criteria

Each of the 160 synthetic data sets were processed using each of the proposed algorithms from Section 4.5 (bar the Nearest Neighbour Interpolation algorithm, as this has been used previously by others for generating maps for wireless positioning algorithms), and the dominant points recorded (yielding a function termed d). The true value at each point in X, $s(x)$, was then compared to the value at that location on the dominant points curve output by the algorithms, $d(x)$. The mean square error (MSE) of all the points in X was calculated, i.e. $\frac{\sum_X (s(x)-d(x))^2}{|X|}$. The mean of the MSEs for each algorithm over all the synthetic data derived from each source curve was then calculated.

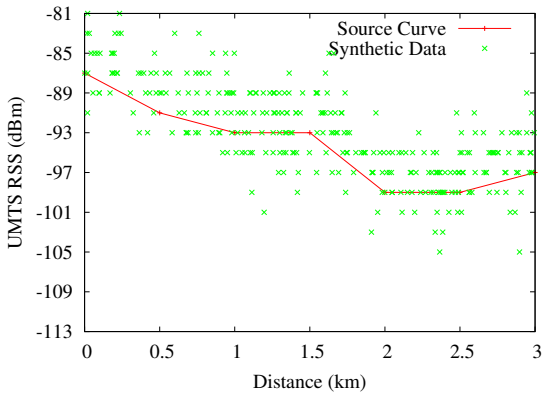


Figure 4.8: Source curve with synthetic UMTS data generated from it.

In a similar fashion, the compression ratio (CR) was also evaluated for each algorithm. This metric is commonly used to evaluate dominant point algorithms' ability to approximate an input shape with as few points as possible. It is calculated by dividing the number of dominant points outputted by each algorithm by the number of synthetic data input points (and hence should be < 1). This provides an indication of what compression has been achieved in the representation. To a certain extent, there will exist a trade-off between how compact the result is and how accurate the predictions that can be made using it are.

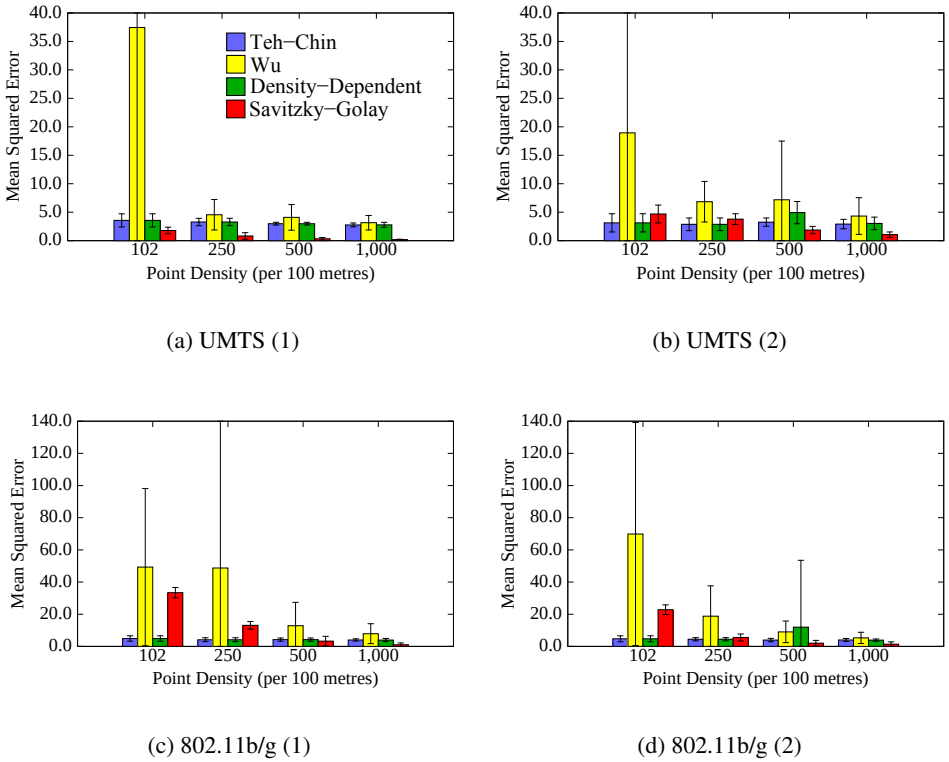
4.7.3 Simulation Results

The results of using the synthetic data are shown in Figures 4.9 and 4.10. Several conclusions can be drawn from them.

Given that the standard deviation of the noise added to the source curve was 3 dBm for UMTS and 3.5 dB 802.11b/g, MSEs of less than 4 dBm (UMTS) or 5 dBm (802.11b/g) suggest that the approximation algorithms perform well in terms of error in correctly representing the curve.

The compression ratios for Wu's and the Savitzky-Golay smoothing algorithms were very low, suggesting very compact representations. Compression ratios decreased with increasing input point density, showing that the algorithms performed well on large quantities of data. Compression by more than a factor 50 was possible: important for large data sets.

Wu's algorithm had poor MSE at low point densities. This is most probably due the fact that at low densities the regions of support of neighbouring points are unlikely to be correlated, which is an assumption made by the algorithm. However, at high densities, this algorithm's MSE was comparable to that of the others, and the standard deviation of its MSE was also much reduced.



(a) UMTS (1)

(b) UMTS (2)

(c) 802.11b/g (1)

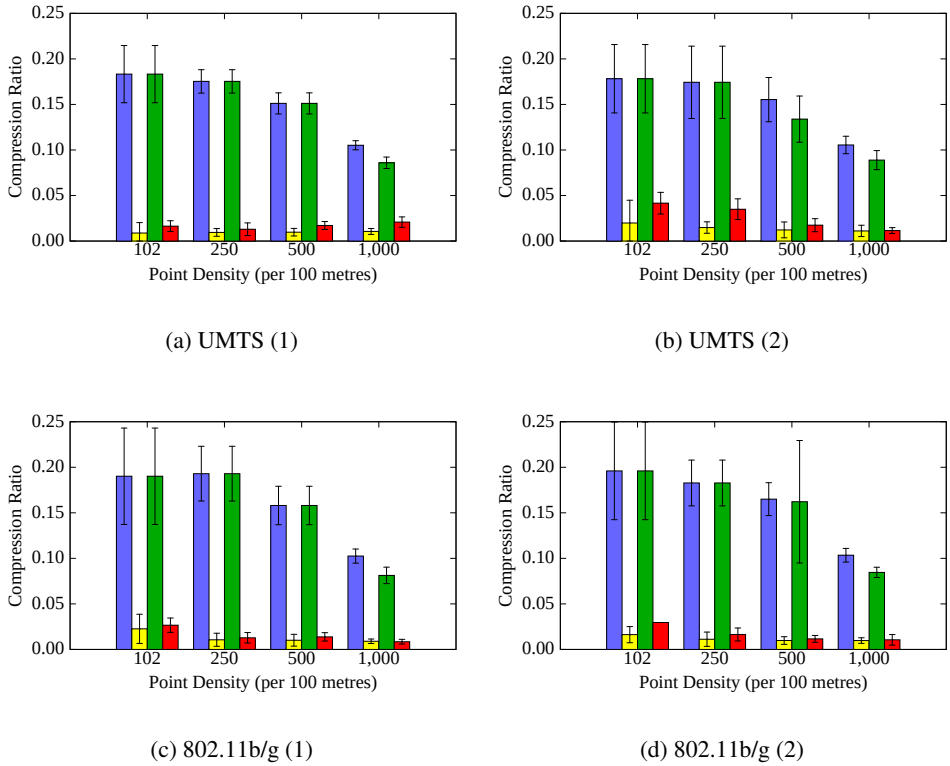
(d) 802.11b/g (2)

Figure 4.9: Comparison of Mean Squared Errors (lower is better) in representation achieved at different point densities, by algorithm, of synthetic data generated from two different source curves for each of UMTS and 802.11b/g.

Savitzky-Golay smoothing had consistently low MSE and CR. The results show that (on synthetic data at least), this algorithm performed consistently well.

The Teh-Chin and Density-Dependent algorithms had consistently poor CRs. At low point densities this was offset by their better MSE than Wu's algorithm. However, at high densities, such high CRs meant Wu's algorithm was (overall) more suitable.

At densities lower than 1000 points per 100 metres, the Density-Dependent algorithm had an MSE comparable to the Teh-Chin algorithm, (and similarly high CRs). At higher densities it retained its very good MSE, whilst achieving CRs lower than those of the Teh-Chin algorithm. Therefore, the Density-Dependent algorithm combined the best of the Teh-Chin and Wu's algorithms.



(a) UMTS (1)

(b) UMTS (2)

(c) 802.11b/g (1)

(d) 802.11b/g (2)

Figure 4.10: Comparison of Compression Ratios (lower is better) achieved at different point densities, by algorithm, of synthetic data generated from two different source curves for each of UMTS and 802.11b/g. Key as for Figure 4.9.

4.7.4 Parameter Optimisation

In addition to the above, experiments were also conducted to ascertain the best value of the k-cosine threshold that should be used. The distribution of k-cosines over the input data is surprisingly non-uniform. Most values are either very close to 1 (implying an angle of close to zero), or close to 0 (implying a right angle). This distribution (particularly at high densities) is due to to the input points being relatively close together, and hence with noisy data the angles will be very sharp.

Because of this quite bimodal distribution of cosine curvatures, it was found that the MSE and CR performance of the dominant point algorithms as the cosine curvature discard threshold was varied between $-1 < \mu < 1$ was a step function; the discontinuity occurred at zero, i.e. when the angle is 90 degrees. At $\mu = -1$ or $\mu = 1$, both the MSE and CR were very high, reflecting that at these discard thresholds, nearly all and none, respectively, of the input points would be discarded. A value of $\mu = -0.9$ was chosen for regions of low point density, in order that only

those points with very large angles would be discarded, as they were unlikely to be important. Points with smaller angles were retained. In contrast, at high point densities, $\mu = -0.1$, and the high discard threshold, μ_2, was chosen to be 0.1, to provide a degree of smoothing of the input data, given that there are large amounts at this density.

4.8 Experimental Evaluation

Having shown that the proposed algorithms perform well on synthetic data, (where their results could be compared against a known value) and optimised their parameters, their performance was then tested on real data. Each algorithm was executed on the corpus of data collected by the Sentient Van to generate a coverage map. The raw data and the resulting coverage maps are shown in Figures 4.11, 4.12 and 4.13. Sensor records from several randomly selected journeys that were *not* in the input corpus were then used in order to evaluate how accurate the predictions made by the coverage map were when compared to the real RSS values experienced on the sample journeys. The space-efficiency of the resulting extents was also analysed.

For each sample journey, each input tuple of 2-D position and sensor value $(\mathbf{l}_i, v_i)$ was snapped to the closest point on the relevant road's centre line, becoming (x_i, v_i). The database was then queried for the coverage map's stored value s_i at length x_i along the road. For all the input points, $d_i = v_i - s_i$ was calculated, as well as the mean and standard deviation of those differences. Each algorithm's extents were evaluated using each sample journey, in order to compare their accuracy. An example of a single journey evaluation for one algorithm over a single road is shown in Figure 4.14.

The two metrics that are important in evaluating the performance of coverage mapping algorithms are the difference between predicted and actual values, and the space-efficiency of the extents. These are considered in turn.

4.8.1 Prediction Error

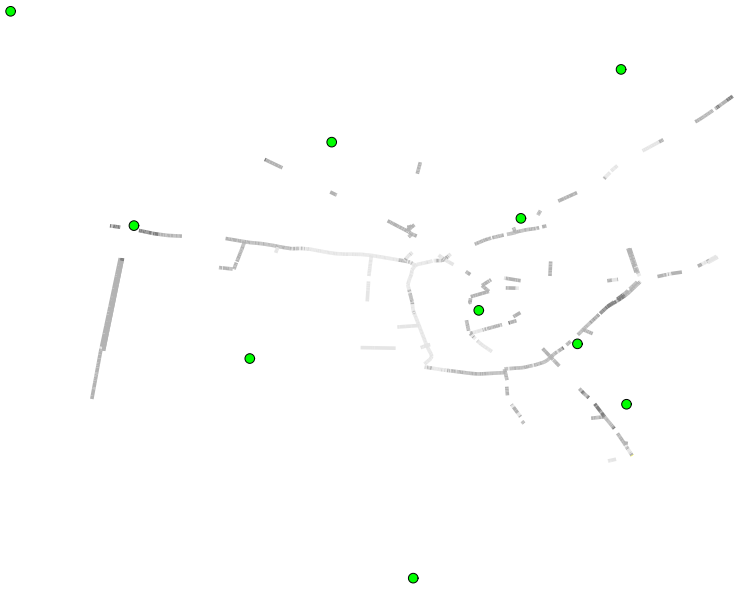
In order to be successful, a coverage map must reliably predict upcoming coverage. The difference between the predicted and obtained values will in part be due to the natural variation in RSS values, as explained in Section 3.2.

Tables 4.1 and 4.2 show how the different algorithms compare for UMTS and 802.11b/g, whilst Figures 4.16(a) and 4.16(c) are the corresponding CDFs. These show that for UMTS prediction, the Density-Dependent and Wu's algorithms performed best, with the Savitzky-Golay smoothing algorithm also having a low prediction error (90% confidence interval of 12 dBm). For 802.11b/g, the Savitzky-Golay algorithm was by far the most accurate, with a 90% CI of only 10.40 dBm.

(a) Raw Data (683,891 points)

(b) Nearest Neighbour (1396)

Figure 4.11: Raw UMTS RSS data and the resulting coverage map, with the number of extents generated.

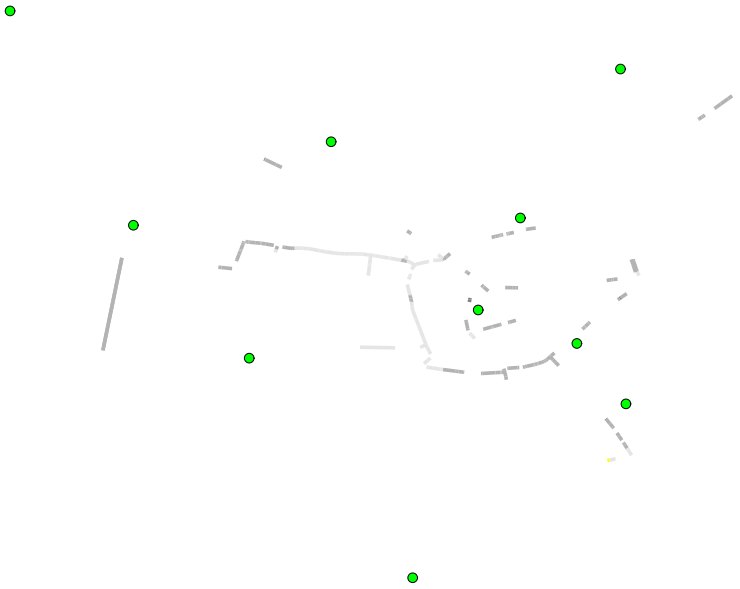


(a) Wu (1552)

(b) Teh-Chin (374)

Figure 4.12: UMTS coverage maps, with the number of extents generated.

(a) Density-Dependent (1204)



(b) Savitzky-Golay (314)

Figure 4.13: UMTS coverage maps, with the number of extents generated.

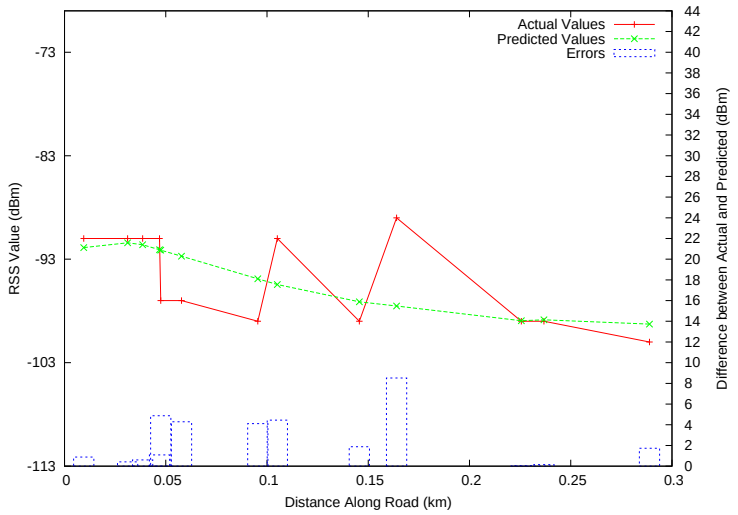


Figure 4.14: Comparison of predicted and actual sample UMTS values from a single journey for a single road link, using Savitzky-Golay smoothing followed by the density-dependent dominant point detection smoothing algorithm.

Figures 4.15(a) and 4.15(b) show the spread of prediction errors. Significantly, the Savitzky-Golay algorithm had only one outlier (circles on the graph) for UMTS prediction errors, showing that its predictions were consistently good. The Density-Dependent and Wu's algorithms did have outliers, suggesting that their performance was sometimes very poor.

An important question is whether these values are significant, e.g. does a 90% confidence interval of 12.00 dBm impact a user far more than one of 14.00 dBm? Table 3.1 shows that for UMTS the relationship between TCP throughput and RSS does not appear to be linear: errors in RSS prediction will be more significant in areas of poor coverage. It is posited that this is because in areas of poor coverage packet losses will be more frequent, each of which will cause TCP's congestion window to fall to near zero. Hence, the window size will never be allowed to increase to large values that would allow high throughputs.

Given this, it is estimated that for RSS values below -90 dBm, an error of 1 dBm is approximately equivalent to 40 Kb/s, whilst the same error at an RSS above -90 dBm would be far less (7 Kb/s). Hence, in areas of poor coverage, a 90% confidence interval of 12.00 dBm translates into a TCP estimate that is approximately 80 Kb/s more accurate than an estimate made with a C.I. of 14.00 dBm. Hence, the difference in algorithm prediction performance is significant for an end user.

Algorithm	$\overline{d}$	σ_d	90% C.I.	$\overline{\lvert d \rvert}$	Tests
Nearest Neighbour	-9.64	4.40	14.64	9.74	748
Teh-Chin	-7.94	4.34	13.30	8.20	748
Wu	-7.72	3.84	12.00	7.92	748
Density-Dependent	-6.86	4.40	12.00	7.08	748
Savitzky-Golay	-7.90	3.56	12.26	8.02	748

Table 4.1: Prediction errors for UMTS (d), all in dBm.

Algorithm	$\overline{d}$	σ_d	90% C.I.	$\overline{\lvert d \rvert}$	Tests
Nearest Neighbour	-7.16	4.86	13.00	7.39	83
Teh-Chin	-5.87	6.41	13.61	7.34	79
Wu	-4.65	6.61	13.57	6.44	65
Density-Dependent	-7.71	6.32	14.00	8.91	59
Savitzky-Golay	-5.07	4.39	10.40	5.71	72

Table 4.2: Prediction errors for 802.1b/g (d), all in dBm.

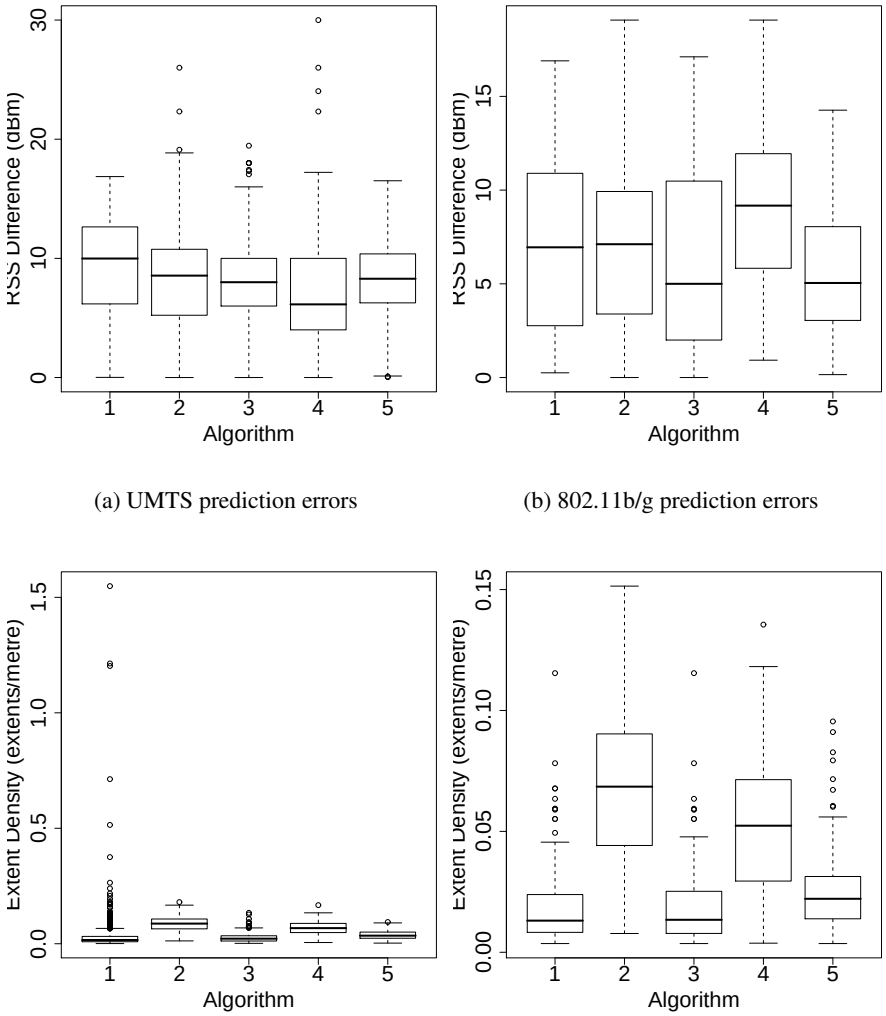
Overall, the algorithms' performance in the very worst case would be an error of 480 Kb/s (poor coverage), or 84 Kb/s (good coverage). These are acceptably low compared to the maximum throughputs achievable, and hence show the utility of the coverage maps generated by the proposed algorithms.

Similarly, it is estimated that a worst case error of 10.00 dBm for 802.11g would correspond to a throughput difference of 4-5 Mb/s. Whilst this is a large value, it should be born in mind that the maximum TCP throughput achievable with 802.11g is 20 Mb/s. Hence, a user will still derive utility from a prediction that is subject to such error.

Overall, it can be concluded that Savitzky-Golay smoothing followed by the an application of the density-dependent dominant points algorithm, performs best, as it combines a low 90% confidence interval in prediction errors for both UMTS and 802.11b/g with few severe prediction errors (outliers).

4.8.2 Extent Density

Ideally, the proposed algorithms should produce as few extents as possible per unit length of road (i.e. a low extent density, ED), in order that the coverage database distributed to a vehicle be small and can be efficiently queried. Tables 4.3 and 4.4 show how the Nearest Neighbour and Wu's algorithms performed well, with few extents produced per metre of road for both UMTS and 802.11b/g. Meanwhile, the Density-Dependent and Teh-Chin's algorithm performed poorly (a low value of ED^{-1} shows how few metres each extent covers), with the Savitzky-Golay algorithm being between these two groups, as shown in the CDFs in Figures 4.16(b)



(a) UMTS prediction errors

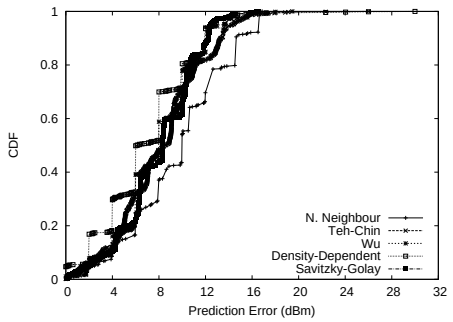
(b) 802.11b/g prediction errors

(c) UMTS extent densities

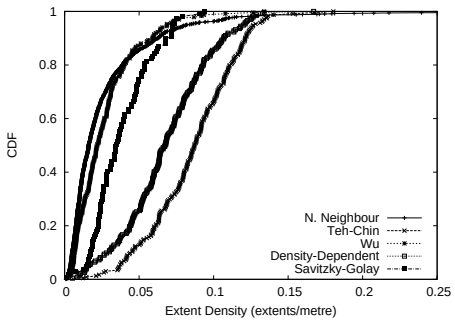
(d) 802.11b/g extent densities

Figure 4.15: Box plots of prediction errors and extent densities by algorithm (Key: 1 Nearest Neighbour, 2 Teh-Chin, 3 Wu, 4 Density-dependent, 5 Savitzky-Golay).

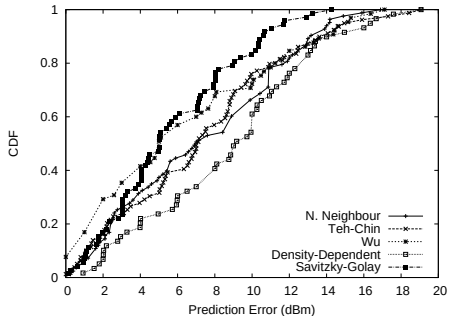
and 4.16(d). Figures 4.15(c) and 4.15(d) show the distributions of extent densities. Significantly, for UMTS the Nearest Neighbour algorithm had several outliers which are indicative of occasional very poor performance (many extents generated per metre). In contrast, Wu's and the Savitzky-Golay algorithm did not have significant outliers. These three algorithms had similar distributions for 802.11b/g.



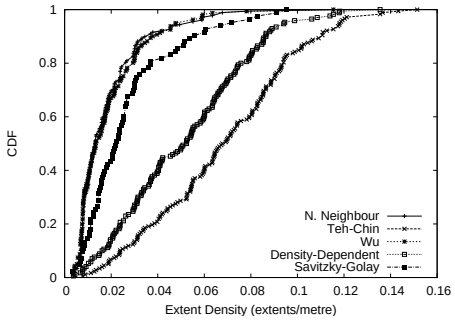
(a) UMTS prediction errors



(b) UMTS extent densities



(c) 802.11b/g prediction errors



(d) 802.11b/g extent densities

Figure 4.16: CDFs of prediction error and extent density. For clarity, the UMTS extent densities x-axis has been truncated.

Given the above, it can be concluded that Wu's algorithm, and Savitzky-Golay smoothing followed by an application of the dynamic density-dependent algorithm, perform well as regards the number of extents generated per metre, and hence are space-efficient. As an example, a typical WiFi hotspot covering 200 metres of road would require only 6 extents in order to represent its coverage,

Overall, it can be concluded that the Savitzky-Golay smoothing followed by an application of the dynamic density-dependent algorithm performs best out of the algorithms proposed, given its low prediction error and good space efficiency. In addition, its benefits over the Nearest Neighbour interpolation algorithm that has been used for coverage mapping efforts in the past are also important.

Algorithm	ED	ED^{-1}	Num. of Roads
Nearest Neighbour	0.030	33.409	1380
Teh-Chin	0.086	11.607	234
Wu	0.028	36.362	234
Density-Dependent	0.068	14.615	234
Savitzky-Golay	0.039	25.756	107

Table 4.3: Mean Extent Density (ED) (extents/m) for UMTS.

Algorithm	ED	ED^{-1}	Num. of Roads
Nearest Neighbour	0.019	53.383	178
Teh-Chin	0.068	14.610	178
Wu	0.019	52.245	161
Density-Dependent	0.052	19.357	170
Savitzky-Golay	0.027	36.577	83

Table 4.4: Mean Extent Density (ED) (extents/m) for 802.11b/g.

4.9 Scalability

4.9.1 Running Time

The computing resources required to produce coverage maps are not onerous. The proposed algorithms were tested on a Pentium IV 3.2 GHz processor with 1 GB of RAM. The system analysed 2,444 roads in the Cambridge area, finding 1,380 roads that had a enough RSS data points to construct a coverage map, and 115 with one or more 802.11b/g networks with the minimum number of points necessary.

The five algorithms were each run on each candidate road, processing a total of over 765,000 UMTS and over 1.2 million 802.11b/g data points. 5,879 UMTS extents and 2,396 802.11b/g extents were generated and added to the database. The entire process took 4,714 seconds. The running time includes the printing of a significant amount of debug output, and therefore could be further decreased. In addition, it should be noted that this figure involves processing each road up to 5 times, whereas in a real deployment only one algorithm would be used.

4.9.2 Mapping Only Useful APs

Whilst this work has presented and evaluated a coverage map for a single provider's cellular network, it has also been indicated that the utility of coverage maps will be greatest when they include information on networks with much smaller coverage areas, such as those of the many WiFi access points (APs) found in cities today. Both this, and work by others, has shown that the number of such APs is very

large, perhaps thousands per city. Whilst a coverage map could be made to include coverage information for each of these APs, many of them will be privately owned, i.e. the majority of users will not be able to utilise them. In addition, many of them will not have sufficient coverage to make their usage by a vehicle worthwhile. Hence, a coverage map only need include those APs that could be useful. This might include all the hotspots for a particular provider that a user has a subscription to, or all those belonging to a community WiFi scheme such as Fon[6].

This "layered" approach, where each provider's hotspots are available as a separately downloadable component to the coverage map, implies that the quantity of storage required for coverage maps on a device will be comparatively low. Thus, whilst coverage maps *can* be extended to mapping all APs in a city, it is likely that each user will pick a subset. A full treatment of the complexity of the directed graphs resulting from the overlapping coverage maps of multiple APs can be found in Section 5.4.

4.9.3 Distributed Computation

It is important to distinguish between the (very large) corpus of RSS data that is collected and the coverage map that is produced from it. The RSS data can be collected in a distributed fashion, then uploaded to a central server. The data is processed using the algorithms described in this Chapter, resulting in a coverage map that is orders of magnitude smaller in size. As more RSS data is collected over time, the storage requirements of the coverage map will not increase significantly, as much of the additional data will serve to reinforce what is already known (and therefore new extents will not be created, but rather old ones will be updated).

In addition, there is no need for a single server to perform all the processing of the data. Because the computation required for each road is independent of the work to be done on any other road, this means that there is significant scope for parallelism. Hence, coverage map generation could easily take place on a regional basis, rather than requiring a nationalised processing centre. Evidently, such an approach would also significantly reduce the quantities of data that would need to be transmitted over large distances.

4.9.4 Impact of Large Numbers of Users

Were coverage maps to become widely used, it is likely that in congested areas a large number of vehicles would all attempt to connect to the same "best" AP, as recommended by the map. This type of heavy use would impact the service level that could be achieved by the AP, and hence differ (probably significantly) from the service level that the coverage map had predicted for it, thus rendering the basic

[6] `http://www.fon.com/`

coverage map useless. A similar problem occurs with satellite navigation devices that recommend using a minor road to a large number of drivers.

However, in the same way that real-time traffic congestion information can now be used to augment personal navigation devices, real-time information concerning network performance is also possible. This could be used to modify the coverage map in order that the previously optimal AP was weighted in proportion to its actual service. In addition, were an AP's service to be always congested, this would be reflected in any performance (rather than only RSS) statistics collected by users, and could be fed back into the production of the coverage maps that are downloaded to vehicles. Such an approach will only become necessary when large numbers of users begin to take advantage of coverage mapping. At that point, there will exist enough of a critical mass to realistically enable real-time reporting of service levels. In addition, network infrastructure is well-placed to inform a coverage mapping provider, or indeed nearby vehicles, of its current service levels, thus providing another avenue for the reporting and distribution of such data.

Such congestion problems are likely to be more of an issue for protocols that use distributed (rather than centralised) medium access control mechanisms. Cellular and WiMax networks will therefore degrade more gracefully than 802.11x deployments. Fortunately, distributed access control technologies have small coverage areas (hundreds of metres in radius). When considering a road running through the coverage of a typical WiFi AP, a reasonable estimate of the length covered (in a city) is 400 metres. Assuming five metres of space per vehicle, and that 50% of them wish to carry out data transfer at once, this equates to a maximum of 40 clients. Hence, even if such a situation were to arise, the service achieved by an 802.11g AP is likely to be more than 1 Mbit/s per client, even assuming that all of the clients wish to use the same AP for the *entirety* of its range. This seems unlikely given the density of APs already deployed in cities.

4.9.5 Extents Versus Boundaries

One previous approach to coverage mapping has sought to determine the *boundaries* of service of APs [152], whilst other research uses the idea of a boundary base station to determine when handovers should occur [205]. It is therefore reasonable to ask whether the accuracy of prediction that can be obtained by using an extents model is necessary. Given that the majority of wireless technologies now use adaptive modulation schemes that depend on prevailing RSS, clearly there will be graduations of service within one base station's coverage area. As higher order modulation schemes are deployed (e.g. HSDPA as compared to basic UMTS) there will be more graduations in service, and the highest throughput service areas will be ever smaller. Hence, the difference between the highest and lowest throughputs achievable when using a particular base station will be far greater, *and* throughput will be even more dependent on location. Thus, whilst coverage boundaries are important, the author believes extents are a more future-proof model.

4.10 Sensitivity to Change

Like any form of map, coverage maps will need to be frequently updated in order to reflect the latest changes in network conditions. Clearly the frequency of update will depend on the frequency of data collection, which in turn is dependent on the number of vehicles that act as sensor platforms ("sensors"), and their degree of mobility: equipping a fleet of delivery vehicles or taxis with logging equipment would be far more effective than several private cars. One advantage of using the vehicles themselves as sensors is that the roads on which there are most users (and hence the greatest demand for accurate coverage maps) will also be those roads on which there is likely to be the greatest sensor density, and hence the most up-to-date maps.

A key question concerns the value of the minimum frequency of measurement updates should take. The data collected by the Sentient Van for Cambridge shows that cellular network deployments appear to be static over long periods of time (see Section 3.2.2), as might be expected, given the cost of installing new base stations. In contrast, user-managed wireless LANs are likely to (dis)appear much more frequently. As a general rule, it is likely that the more investment required in a network deployment, and the greater its geographical coverage, the less frequently its coverage will change.

Until recently, the billing and authentication systems required for public access to a network were confined to high-cost deployments, and hence these networks were unlikely to change frequently. However, the rise of community WiFi networks, particularly British Telecom's strategy of encouraging its broadband subscribers in the UK to join Fon's community network, is changing this. Now private network infrastructure also forms part of a public wide-area network, but one which has a high degree of churn in its constituents. Such developments require more frequent map updates.

Analysing three years of data from the Sentient Van shows that many private networks recorded in the first year of collection are not present in the third year. This is due, in part, to the high population churn of a university city, but also the rate of replacement of consumer devices. Updates are therefore likely to be required at least once per year, a frequency which is easily achievable even with a very low penetration of sensor-equipped vehicles. The problem is akin to the well known (geographical) map updates that firms such as Tele Atlas and Navteq have to conduct, in part using notifications from local governments, and for the remainder driving many thousands of kilometres in their own probe vehicles. Similar notification mechanisms could apply to new deployments or decommissioning by wireless infrastructure providers (in whose interest up-to-date coverage maps would be), whilst dedicated probe vehicles could be replaced with a more community-oriented approach, where data collected by private individuals is used by the system.

Three separate scenarios are possible that would cause an update to be required:

- **Addition of an AP:** probe vehicles reporting RSS readings from a new AP would be the main source for additions to coverage maps. To target where dedicated (as opposed to volunteer private users') vehicles should drive, notifications from commercial providers of infrastructure deployments could be used to good effect.

- **Removal of an AP:** detection of the *lack* of an AP is somewhat more difficult than detecting its *presence*. In addition to any notification systems, removals could be detected by maintaining a timestamp concerning when the most recent data has been received for each road, and a similar timestamp for each extent. When the coverage map algorithms are run, if a particular extent's timestamp does not change, but its road's does, this would indicate that that wireless network was no longer present. Several such incidents would mean that the network could be removed from the coverage map.

- **Modification of coverage of an existing AP:** changes in the environment, such as the construction of a new building, will influence wireless coverage. Probe vehicles are most important in this task. However, probe vehicle targetting can again be achieved if *users* of coverage maps report not *all* sensor data, but instead where a coverage map's prediction was significantly different from what was experienced. Such feedback mechanisms are already beginning to emerge for road mapping, such as Tele Atlas' Map Insight[7].

Assuming that updates are performed with sufficient frequency, how the coverage maps themselves are updated using the new data must also be examined. One simple approach is to assign weights to RSS values which exponentially decrease with their age. This is simple in the case of the nearest neighbour interpolation and Savitzky-Golay algorithms, which already use weights, and could be incorporated into the dominant point algorithms by modifying the value of κ, depending on the age of the point being considered for discard. The weightings used would depend on how sensitive the entity generating the coverage maps wished them to be to changes. Clearly, a map that immediately reflects a change in coverage is likely to yield greater prediction errors, due to placing a very high weight on the very newest readings, which might include a small number of outlying values. More conservative strategies will take longer to update, but will yield more accurate maps in the long term. Linked to this is the possibility of notification by providers of infrastructure changes, which would then allow weightings for a particular area to be changed to favour more recent data.

[7] `http://mapinsight.teleatlas.com/`

4.11 Distribution

Distributing coverage maps gives rise to the same problems, which succumb to similar solutions, as the distribution of digital road maps, as used in personal navigation devices. We have already seen the transition to regular updates sent out on removable media by map makers, and over the air update systems are beginning to become realistic. Meanwhile, many mapping services are now available online, and hence can be consulted by a device whilst on the move. Updates to such services are instantly visible to their many users as soon as the new data is deployed on the provider's server. Thus, distribution becomes an even more simple problem as services migrate from dedicated devices into "the cloud".

In terms of over the air updates, these can be fetched by background transfer. Detailed coverage maps for particular areas need not be fetched until a user is nearby that area, whilst low resolution (e.g. showing only the user's main cellular provider's base stations, rather than individual WiFi APs) coverage maps could be provided for other areas for the purposes of large-scale route planning.

Because coverage maps as proposed in this work have been designed to map onto the directed graph paradigm, they could easily be included with road map updates, and indeed included on personal navigation devices without significant modification. Hence, distribution is not as significant an issue as it might first have appeared.

4.12 Applicability to Other Data Types

Whilst this Chapter has shown the suitability of the proposed algorithms for wireless signal strength coverage mapping, they can also be applied to other data types. Several of these are given below. They illustrate how very large datasets can be represented compactly yet accurately using extents.

4.12.1 Vehicle Speeds

Many of the satellite navigation units now being sold include databases with typical traffic speeds on each road at various times over the day. This allows routing algorithms to take better account of congestion. The algorithms proposed in this Chapter could enable such databases to provide more accurate speed data whilst still occupying a similar amount of storage. This is because extents are not a uniform length, but instead vary depending on how far along a road the raw values are constant for. Hence, the same number of extents could show more accurately the congestion at junctions versus the otherwise clear nature of a road. Figure 4.17 shows the map created using 2.1 million GPS speed readings from the Sentient Van. Of particular note are the red portions very close to junctions, demonstrating the success of the algorithms at picking out short but sharp peaks in the raw data.

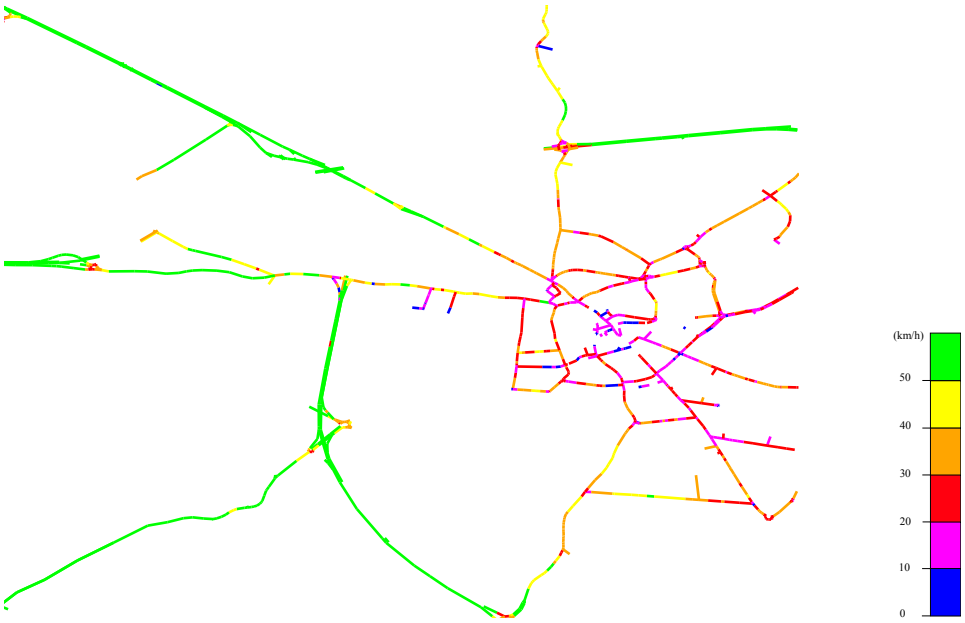


Figure 4.17: Map of vehicle speeds on Cambridge roads (Nearest Neighbour interpolation).

4.12.2 Carbon Dioxide Concentration

A further data set collected by the Sentient Van consists of Carbon Dioxide concentration readings. These are in parts per million, and are for air taken from an inlet sited on the roof at the rear of the vehicle. The data set consists of 2.8 million readings. The sensor has not been calibrated, and hence the readings on the map illustrate trends rather than absolute values.

4.12.3 Ambient Noise

Courtesy of the MESSAGE project [129], data concerning ambient noise on Cambridge roads was also processed. This was collected by cyclists using the microphone portion of a hands-free kit connected to a GPS-receiver-equipped mobile phone. The total number of readings processed was over 58,000. The resulting map is shown in Figure 4.19, with the units being arbitrary (higher corresponds to louder).

Figure 4.18: Map of carbon dioxide concentration on Cambridge roads (Nearest Neighbour interpolation).

4.13 Chapter Summary

This chapter has described how wireless network coverage maps can be created. Existing techniques for creating such maps have been described, such as inverse distance weighting, contour simplification, and propagation simulations. Such methods do not lend themselves well to large quantities of noisy RSS data. By restricting the problem to roads (or other transport corridors), the problem can be approached as one of line simplification. This chapter has presented several novel techniques that apply modified versions of existing algorithms from other fields to this task. By evaluating these algorithms on synthetic data, the optimal parameters for the problem were chosen. The algorithms were then evaluated using real data to ascertain their accuracy of using extents to represent the raw RSS readings, in addition to how space-efficient that representation was. Finally, the algorithms were used to process other types of data, illustrating their general applicability.

Hence, it has been shown that by using these extent generation algorithms, accurate and compact coverage maps of wireless networks can be generated. Such coverage maps are a key part of enabling wireless devices to carry out network selection and inter-network handovers optimally. In Chapter 5, a method of using these coverage maps is presented. Specifically, for a large number of networks, the best sequence of networks to connect to over a geographical route can be calculated. This takes advantage of the extents representation that can be easily converted into a directed graph, allowing shortest path routing algorithms to be applied to the problem.

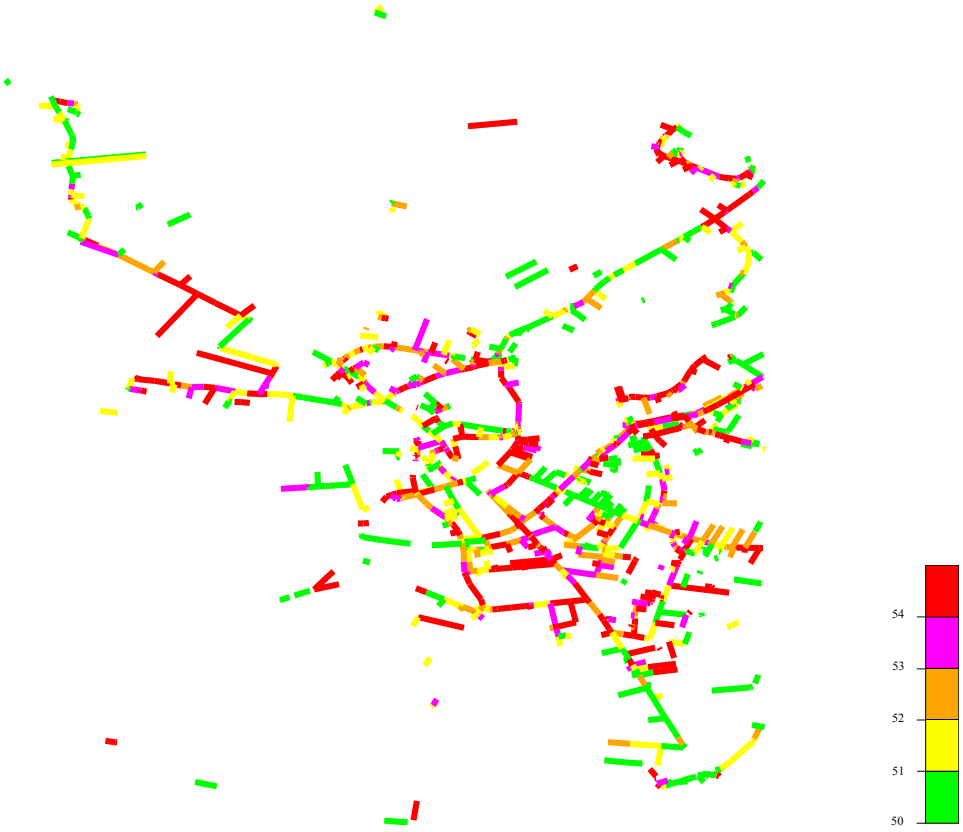


Figure 4.19: Map of ambient noise on Cambridge roads (Nearest Neighbour interpolation).

5

Constructing Multi-Planar Graphs from Coverage Maps

In Chapter 4 algorithms for constructing space-efficient and accurate coverage maps were proposed and evaluated. The key aim of the algorithms was to propose a representation that could accommodate many overlapping networks in a format that could be easily queried. This Chapter proceeds to show how coverage maps' extents can be converted into a multi-planar graph, which can represent not only coverage but also the costs of handovers between different networks. Optimisations are described that enable the graph's complexity to be significantly reduced, and an in depth analysis of that complexity is provided.

5.1 Introduction

As described in Chapter 2, previous work on proactive handover decision algorithms has enabled client devices to prepare for handovers, but has not addressed the optimal moment to hand over at. This Chapter describes a proactive scheme that utilises the coverage maps proposed in Chapter 4. These maps detail all the wireless networks (of all technologies) that are available. Coverage maps can be used not only to predict when a handover is about to take place (e.g. due to a black spot in the coverage of the network currently in use), but also as a parameter utilised by routing (navigation) algorithms. To the best of the author's knowledge there is no previous work that has implemented such an approach.

At first glance, routing using coverage maps appears simple. Currently, most navigation units employ a version of Dijkstra's shortest path algorithm, using a metric that is a combination of the length of each road and its traversal time (based on the road's speed limit). Other metrics may also be included, such as the economic cost of each edge (toll roads being a pertinent example). Unfortunately, this simple augmentation of the existing graph of roads does not work when multiple wireless networks are involved, as will be explained in Section 5.2.

The focus of this Chapter is therefore on how a coverage map is converted into a directed graph. Handovers are represented between the multiple (overlapping) networks of different technologies, using appropriate cost metrics. Using this augmented graph, it is possible to calculate when and to which network a client should

#	h	bt (Mbit)	$\overline{b}$ (Mbit/s)	d (%)	t (s)
1	11	1392.5	2.59	29.1	538.1
2	17	2116.6	4.07	25.3	519.6
3	11	1693.1	3.44	27.7	491.7
4	14	2285.9	4.40	26.6	520.0

Table 5.1: Connectivity statistics for the different routes between two points as shown in Figure 5.1. Number of handovers h, total data transfer bt, mean throughput $\overline{b}$, percentage of time spent disconnected d, and total time to traverse the route t.

hand over in order to achieve the best QoS. This representation also allows the cost/benefit of the handover to be evaluated quantitatively before it needs to be carried out.

The proposed system provides a route that may be geographically distinct from the shortest path, in order that network QoS is maximised. Other implementations of proactive algorithms have concentrated on providing a (not necessarily optimal) handover sequence for a *given* route. The system is novel in that it suggests alternative *routes* that optimise connectivity (as well as taking into account their traversal time). It requires no augmentation of existing network base stations, and therefore can be incrementally deployed, assuming that sufficient RSS readings can be collected to construct the required coverage maps. Finally, it is important to note that the majority of previous work in this field has been carried out using simulation, rather than by using real RSS data from an urban environment. In contrast, the work described here makes use of the large corpus of data recorded by the Sentient Van (Section 3.1), and uses real traces to evaluate the approach.

5.1.1 The Value of QoS-Aware Routing

Figure 5.1 illustrates the need for QoS-aware routing. Four possible routes between the two locations marked with circles are depicted, each generated by the choice routing algorithm [120]. Each route is shown in a different colour, although they overlap in some portions. According to the coverage map for this area, one route (number 4) is calculated as having a traversal time 0.1% longer than another (route 2), but achieving a data transfer of over 20 MB more (after interruptions due to handovers are taken into account) and carrying out three fewer handovers. Were a user to require a particular file to be transferred to their vehicle whilst driving to work, this tiny detour might be precisely what is needed. Table 5.1 compares all the routes in detail, further illustrating the significant differences between them.

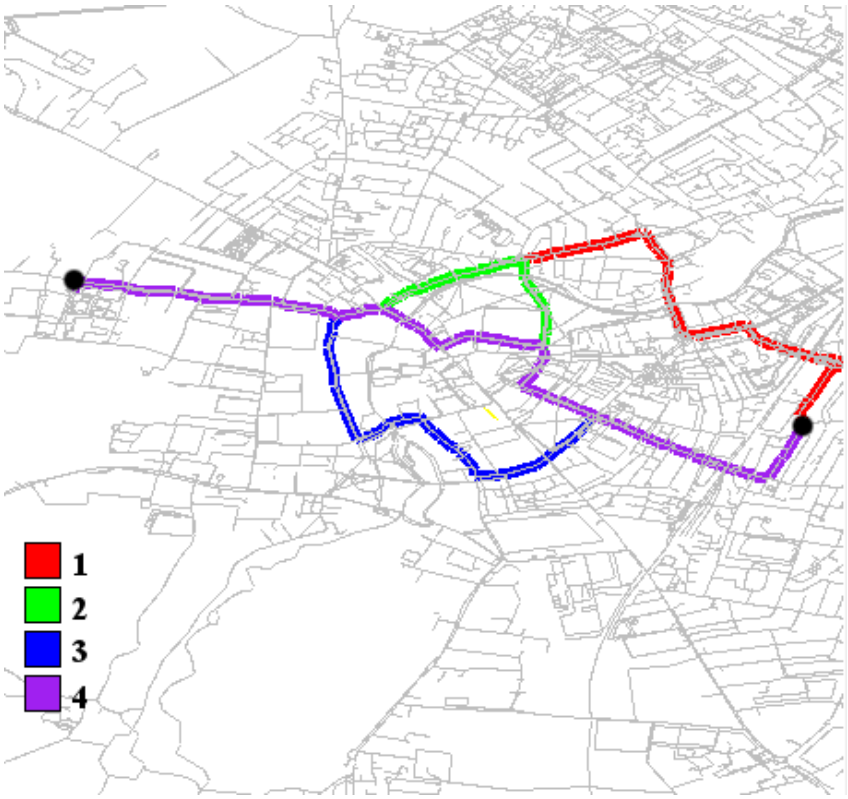


Figure 5.1: Four possible routes between the two points marked by circles [OSM].

5.1.2 Use Cases

In order to justify the applicability of intelligent handover algorithms to existing problems, it is crucial to examine their real-life applications [180]. First, the applications of handover algorithms are split into two problem domains: those where the algorithm is given a predetermined geographical route to optimise network QoS over (e.g. the shortest path); and those where the algorithm is allowed to choose the geographical route *based on* the network QoS. These will be termed *constrained* and *unconstrained*, respectively.

The constrained case is the more familiar of the two, and handover algorithm implementations to date have assumed that the user of a vehicle will specify a particular route, which the algorithm cannot influence. Here, the goal is to select the sequence of networks to be connected to along the route to that provide the best QoS for the user. Three use cases are apparent:

- **Maximise Transfer.** Attempting to maximise the total amount of data that can be transferred over a route. This would be useful for disconnection-

tolerant applications, such as downloading map updates or uploading sensor data that has been cached on the vehicle, as quickly as possible.

- **Minimise Handovers.** Changing the network or network technology a client is connected to involves handover costs. Not only is this in the disconnection time incurred, but also because of transport layer protocols needing to adapt to the new throughput and delay characteristics, and recover from any packet loss. Minimising the number of handovers carried out is therefore important if a stable connection is needed (e.g. for multimedia streaming). If such applications began to be used during the journey, the routing algorithm could be informed of the QoS requirements and automatically re-route to satisfy them, without user interaction.

- **Minimise Disconnections.** Attempting to ensure that as high a percentage of the route as possible has *some* connectivity, though this may not be of particularly high throughput. This is similar to attempting to achieve a particular target throughput over the entire route (see below). However, it differs in that whilst we might wish for a particular target, in many cases even a meagre amount of connectivity is better than none. This is the case for applications where the bit rate can be lowered to give degraded quality, if necessary, e.g. a two-way video call degrading temporarily to audio only.

Increasingly, drivers are placing their trust in satellite navigation devices, which, when given a start/destination pair, will find a "good" route (where the meaning of "good" varies from unit to unit, depending on the weighting each road is given). Drivers then follow the route provided, in many cases not thinking particularly carefully about where it takes them. In addition, higher end navigation units now include traffic congestion updates that cause the generated routes to deviate from the shortest path in terms of distance, thus increasing the complexity of the problem. Therefore, algorithms run by navigation devices, rather than humans, are dictating the geographical routes to be followed. Moreover, given that the problem is only likely to become yet more computationally complex (e.g. the inclusion of time-varying road tolls in the calculation of cost), it is likely that navigation devices will become ubiquitous in vehicles in the near future. Compounded with increasing demand for in-vehicle Internet connectivity, and the complexity of selecting the best network to connect to, the author envisages a fusion of navigation and location-dependent connectivity information. In this case, a handover decision algorithm will be run in an unconstrained manner, i.e. it will be able to influence the geographical route that the navigation device recommends. In addition to the use cases described for the constrained case, two other use cases are of interest:[1]

[1] Other use cases incorporating factors such as financial cost, e.g. finding a route that has the best QoS within a certain budget, are also relevant. This dissertation restricts itself to network QoS and route length.

- **Minimum Throughput.** The requirement for a particular minimum (target) throughput to be present over as much of a route as possible. This would be the case for an application which involved real-time media streaming, such as a voice-over-IP conversation, or streaming a film for passengers to watch.

- **Transfer by Arrival.** Allowing a quota to be set concerning the amount of data which should be transferred by the end of the journey. This is relevant to usage scenarios such as an office worker needing a particular presentation in time for a meeting, or emergency workers obtaining building plans.

For these use cases a route that meets the goals expressed may not be the shortest geographical route. For example, it might be that the shortest route has no connectivity for a small portion, which would not meet a goal of a particular minimum throughput. Hence, in some situations it might be beneficial to generate a route that has a better network QoS. However, this should not come at the price of a route length that is significantly greater than that of the shortest path. As shown in Table 5.1, there is often more than one route of similar length between two points, yet they may have widely differing network characteristics. The goal is therefore to find a route that is similar in length to the shortest path route, but with better network QoS. The remainder of this Chapter describes how the graph structure necessary for such route finding is constructed.

5.2 Coverage as a Graph

For routing between two points in a road network, the standard representation of the road topology as a directed graph is adopted. Here, a graph $G = (N, E)$ is made up of a set of nodes N, and a set of edges E. Each bidirectional road consists of two edges running between two junction nodes, one in each direction. Unidirectional roads are represented by a single edge in the appropriate direction. Each edge e_i runs from node $n_{i,1}$ to node $n_{i,2}$ and has a weighting, w_i. This representation permits for the application of standard shortest path algorithms to find the lowest total weight route. Typically, the edge weights are modified to take into account the expected traversal time, t_i, and length, l_i of the edge, and hence favour routes that are faster rather than shorter.

5.2.1 The Single Network Case

Extending this idea to route selection based on maximal network throughput for a single wireless network, w_i could also be made to incorporate an RSS value. However, this fails to account for the inevitable RSS variations along a road. Hence we incorporate the notion of network coverage into G by adding further nodes and edges, specific to each wireless network, that represent such variation. These

virtual nodes are distinct from (road) *junction* nodes in that they correspond to locations where RSS changes, rather than physical features such as crossroads or roundabouts.

As described in Section 4.5, a coverage map represents each road as a series of segments termed *extents*. Each extent is defined to be a subsection of the road across which the network coverage is constant or varies linearly to a first approximation. It consists of a tuple $\{s_1, s_2, v_1, v_2\}$, where s_1 and s_2 are the start and end of the extent, each given in terms of one-dimensional distance along the road. The terms v_1 and v_2 represent the coverage values at s_1 and s_2 respectively, and coverage varies linearly between these points. For a given wireless network, the set of extents on any road will be non-overlapping[2], and therefore at any point on the road the coverage map will predict a unique value for the coverage.

For each road i, then, there is a set of non-overlapping extents X_i indexed by j, $j \in [1..|X_i|]$, which can be ordered by their start lengths $s_{j,1}$, and which together span the entire length of the road. Extents x_1 and $x_{|X_i|}$ are respectively the first and last extents on this road, and hence $s_{1,1} = 0$ will correspond to the start junction node of the road ($n_{i,1}$) and $s_{|X_i|,2}$ to the end junction node ($n_{i,2}$). To add the extents into the road graph, new virtual nodes are created in the graph, located at the boundaries between the extents. A virtual node $v_{i,k}$ is created at every $s_{k,2}$ where $k \in [1..|X_i| - 1]$. This process is shown in Figure 5.2.

Virtual nodes are connected by new edges, each of which has a weight that is proportional to the length and traversal time of the new edge, *and* the mean RSS expected along it ($(v_{j,1} + v_{j,2})/2$). This new graph can now be used for routing which takes into account the expected throughput that will be achieved[3]. By changing the weightings of the quantities used to generate the values of w_i, it is possible to specify how important connectivity is compared to the length or time of the route.

Kamakaris and Nickerson [127] proposed a system along these lines. A contour map of a *single* network's RSS would be generated, then each road would be compared to the map to see within which contour it fell. The road graph would then be augmented with the appropriate RSS value. However, their proposed system did not take into account what would be done when a road spanned more than one contour. Moreover, the system was not actually implemented, but only proposed as further work.

[2]Given that extents along a road are contiguous, a more compact representation could include only one length and value per extent, with the next extent implying the end distance and value of its predecessor. For clarity of explanation, the less compact form has been retained.

[3]Because the algorithm will prefer lower weightings, the *inverse* of the throughput is incorporated into the weightings, (though in Section 6.1 it is shown that this is not in fact a correct approach).

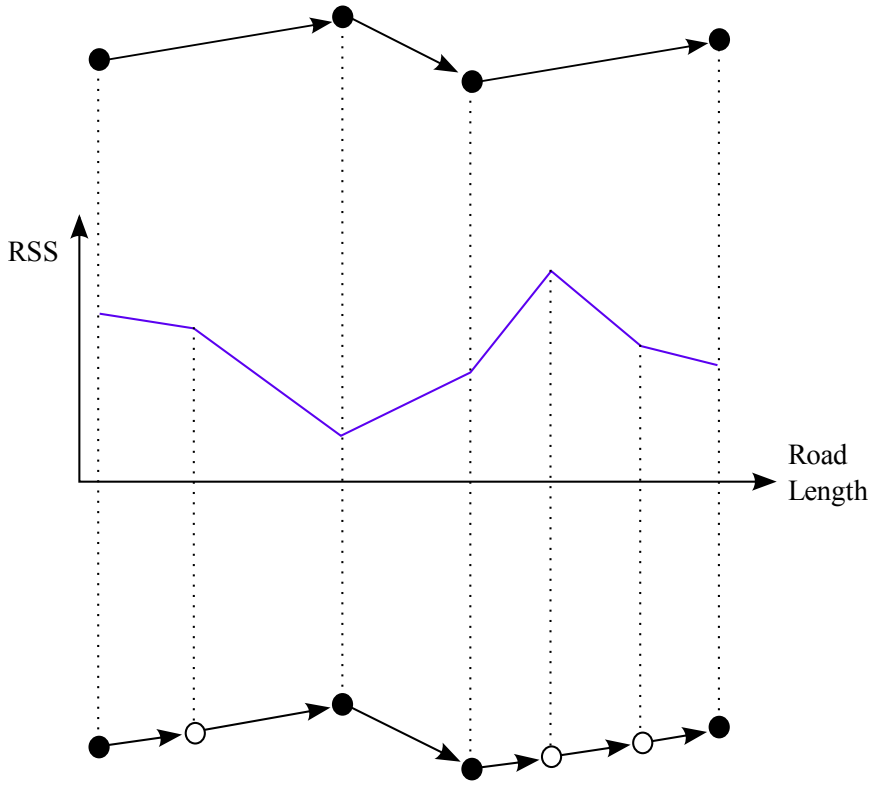


Figure 5.2: The original road network (top) consists of junction nodes (black circles) and directed edges (roads). The coverage map (middle) is a profile of RSS over road length. This is used to add virtual nodes (white circles) to the graph at extent boundaries (bottom).

5.2.2 Inapplicability to Multiple Networks

Whilst the above approach is useful when considering only a single wireless network, it cannot be easily applied to routing using multiple overlapping networks. Figure 5.3 shows an example, where three nodes are connected together by numbered edges. Each edge has a tuple (t, w) which corresponds to the network type and the weight of that edge in the graph. Edges 1 and 4 have the same type. If we assume $w_1 < w_2$ (i.e. network type t_1 is faster, or will yield a higher throughput), a standard greedy routing algorithm will choose edge 1 to move from A to B. If $w_3 < w_4$, the same argument sees the algorithm select edge 3 to move from B to C. Since edges 1 and 3 have different types, a vertical handover between network technologies will be required at B. However, a handover is not an instantaneous event, and will disrupt the QoS at and around B. If w_3 is only slightly less than w_4 it is preferable for a routing algorithm to avoid this disruption by selecting edges 1

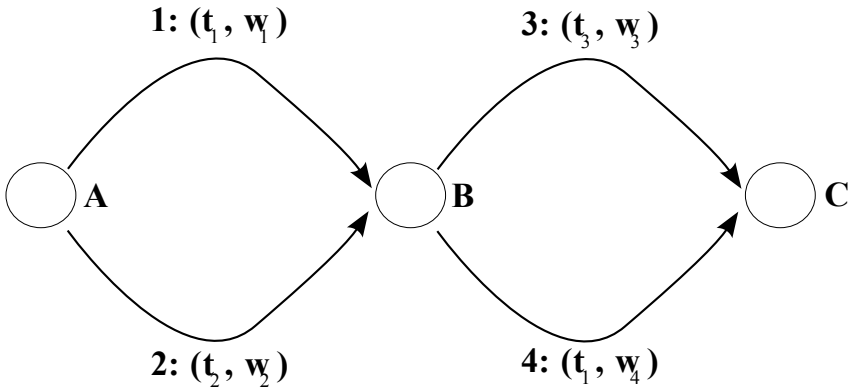


Figure 5.3: Illustration of how multiple network types in one graph does not achieve the goal of QoS-aware routing.

and 4, resulting in no handover. Unfortunately, the model outlined thus far cannot achieve this goal.

One solution is to modify the weight of edge 3 to take into account the "cost" associated with a handover (be it packets requiring retransmission, period of time disconnected, or another similar measure). Any such modification would be dependent on the path traversed to that point (i.e. w_3 is dependent on the path taken from A to B). Worse, there is potential for reverse dependencies, which could occur in this example if $w_2 < w_1$. In this case, edge 2 would be chosen to reach to B, before discovering that edge 1 was a better choice, since it is possible to use the same network for the entirety of the journey A to C. The solution to such problems requires back-tracking as the graph is traversed. Potentially, such back-tracking might be all the way to the start node of the route. Traditional routing algorithms such as Dijkstra's do not provide back-tracking capabilities, and hence will not provide the handover-aware route that is required.

The contribution of this Chapter is to provide a solution that allows a route that is good both in terms of traversal time and connectivity to be found efficiently in the presence of multiple overlapping wireless networks. The proposed method takes handover costs into account, and provides the sequence of handovers that should be performed when traversing the identified route.

5.3 Routing for Handovers

In order to represent extents belonging to multiple wireless networks over a road topology, without being subject to the optimisation problem discussed above, the

concept of a multi-planar graph[4] is used. Here, it is useful to keep in mind the analogy of a multi-storey car park. These normally involve multiple parking areas of identical layout built above one another, with separate ramps for ascent and descent interconnecting them. When representing available wireless networks, each "floor" or plane corresponds to a single wireless network's copy of the global road topology. Handovers between networks can be carried out by ascending or descending "ramps", each of which has an associated cost. With this approach a greedy routing algorithm can be used to find a route with a good handover sequence. Therefore, the next Section considers how the multi-planar graph is constructed.

5.3.1 Virtual Nodes

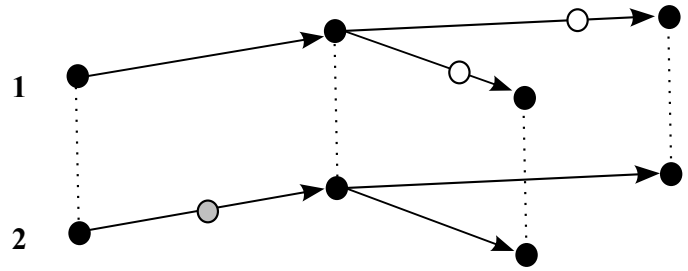
As described in Section 5.2, extents are represented using virtual nodes and edges. Given that an extent represents a region of constant (or linearly varying) RSS along a road, and is likely to be relatively short, the constraint that inter-network handovers will only be performed at extent boundaries is applied. Although not a true physical constraint, it has little impact on the outcome (since a significant, rapid change in RSS will by definition occur at an extent boundary) and greatly decreases the number of "ramps" in the final graph.

Each wireless network or administrative domain is represented as a graph lying in the x-y plane which has the same fundamental layout to the road network, but with additional virtual nodes to model the extents of wireless network coverage. To facilitate transitions, a virtual node in one plane must have a direct counterpart in each of the other planes. Thus, virtual nodes are inserted into each plane to ensure that network transitions are possible at every node. This process is depicted in Figure 5.4.

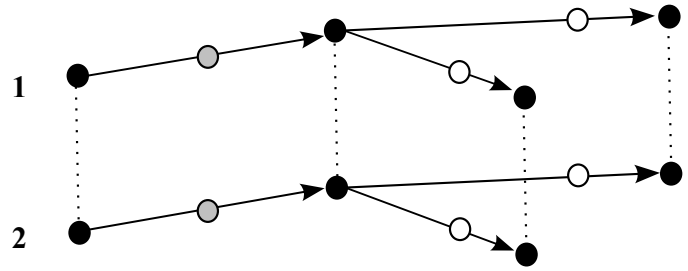
5.3.2 Handover Nodes

There is, however, a further complication. Since handovers are not instantaneous, the vehicle position shifts before the transition between networks is completed. Thus, the position of the virtual node after transition is incorrect if we naïvely transition "vertically" between planes (which fundamentally assumes the handover was instantaneous). In order to model this, *handover* nodes are created that use

[4]Strictly, a planar graph is one which has no edges that cross each other. A road network may therefore not be planar in this strict sense. However, the notion of planes is used to distinguish between multiple copies of the original road network that make up the overall graph.



(a) Original road network



(b) After virtual node addition

Figure 5.4: Adding virtual nodes to all network graphs. Black nodes are junction nodes; white and grey nodes are virtual nodes corresponding to extent boundaries for network 1 and network 2 respectively. The upper diagram is before node addition, the lower shows how nodes are replicated across the different levels.

the previously measured handover time and the maximum permitted speed of the vehicle to estimate the furthest that could be travelled during the handover. The *maximum* achievable distance must be used, since the goal is to model how much of the destination extent remains for "use" after the handover. This latter quantity can be underestimated (the client will simply obtain connectivity for longer than expected, thus deriving a benefit), but should not be overestimated.

Before proceeding, it should be noted that although the graphs depicted in this Chapter show one network vertically above the other, this is only for illustration. Inter-network handovers are permitted between any two networks, and hence when creating handover connections in the graph, every node is connected to a handover node in every other network. For clarity, the examples show only two networks, whereas the actual graphs produced in the system are significantly more complex.

In order to generate the handover nodes and connections between a set of graphs for different wireless networks, $\mathbb{G}$, each network graph $G_k = (N_k, E_k) \in \mathbb{G}$ is considered in turn. For each node $n_{k,i} \in N_k$, the set $C_i = \{n_{x,y} \in \bigcup N_k | x \neq k, y = i\}$ is generated, containing all nodes in $\mathbb{G}$ with the same geographical position as $n_{k,i}$, but that belong to other wireless networks. For each node $c_{x,i} \in C_i$, the edges in E_x that have this node as their source are examined[5]. Each of these edges must have a handover node added to it, as each is a possible end point if a handover is begun at this node. The process begins by looking up the handover time, t_x, from network k to network x. This, in conjunction with the edge's speed limit, enables the calculation of how far along the edge the handover node should be sited (i.e. how far a vehicle can legally travel within t_x). The handover node is inserted into N_k, and E_k is updated to connect the new node into the existing graph. Figure 5.5 shows the process part of the way through, where handover nodes (squares) have been inserted that correspond to two of the existing (circular) nodes.

5.3.3 Complications Due to Graph Cycles

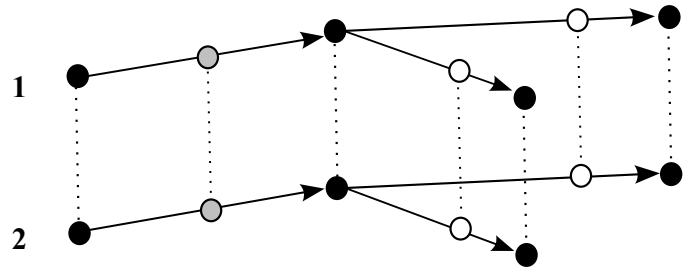
One complication to this process is that the handover time t_x might permit the vehicle to exit the edge altogether. In this case the algorithm follows all possible exit edges and places a handover node on each at the appropriate distance. In an implementation, there is the potential for G_k to contain cycles, and consequently some of the outgoing edges may be too short to permit handover. This can arise because an outgoing edge from $c_{x,i}$ is the start of a path that leads back to $c_{x,i}$, which has a total traversal time of less than t_x. If this occurs, a handover node is not added on that particular path. Another implementation detail is that handover nodes may be assigned that coincide with an existing node. In this case the new handover edge is connected to the existing node, instead of a new node being created.

5.3.4 Adding Handover Edges

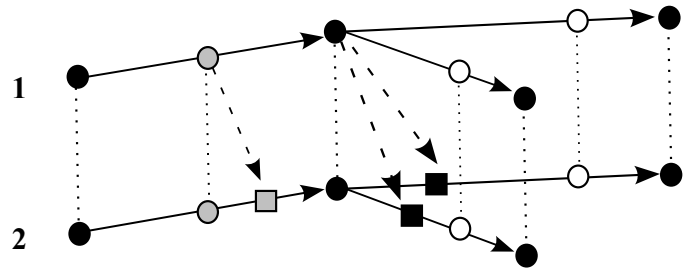
Once handover nodes have been inserted, it is a simple matter to connect up each extent boundary node to its corresponding handover nodes, with an appropriate time and distance cost. Throughput is assumed to be zero during the handover, implying complete disconnection. In this work, soft handovers (where a mobile terminal can listen on two network interfaces at once, and hence always remain connected) are assumed *not* to be in use, although adapting the proposed scheme to handle such situations should be simple.[6]

[5] Those with $c_{x,i}$ as their destination are *not* examined, as a handover connection should not imply that there is path in the opposite direction to the direction of the street.

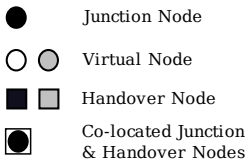
[6] Including soft handovers would involve decreasing the penalty incurred due to handovers to only that of adapting the transport protocol's behaviour to the characteristics of the new link. Other penalties such as the extra energy consumption of multiple radio devices, or increased signal processing load could also be factored into the handover edge cost, if so desired.



(a) Before handover node addition



(b) After handover node addition



(c) Key

Figure 5.5: Adding handover nodes. The first diagram shows the two networks from Figure 5.4, whilst the second diagram shows a selection of the handover nodes that are added. Square nodes represent handover nodes, their colour corresponding to that of the source node to which they are connected.

The final step in creating the multi-planar graph is to add handover connections which have a cost of zero from the start node to all other nodes with the same geographical location. An analogous process is carried out for the end node. This allows the routing algorithm to begin the route on the most appropriate wireless network, rather than there being a handover penalty at the outset because the "wrong" starting plane was selected.

Once the graph is completed, we can use it to find a route between two points in terms of distance, time, and network characteristics. Depending on the weighting function that is chosen, each of these can be made more or less important. Network characteristics such as throughput or aggregate data transfer can be used depending on the application requirements. For example, an interactive gaming application relies on a minimum throughput and minimal disconnects; a film download will likely require the highest aggregate transfer per edge and can tolerate disconnections.

5.4 Graph Complexity

5.4.1 Complexity of Initial Approach

In the approach described in the previous Section it is clear that the number of distinct wireless networks, m, will cause an unaugmented road topology graph $G = (N, E)$ to expand to $G' = (N', E')$ with at least $|N'| = m|N|$ nodes and $|E'| = m|E|$ edges. This conservative estimate assumes only one extent per wireless network that spans the entirety of each edge. In practice, to be of use in predicting non-trivial changes in coverage, extents are normally much shorter than entire road lengths. The exact dimensions of an extent are dependent on the algorithm that is used to process the raw RSS data, and the parameters that algorithm is given. The trade-off between average extent length (and hence how compactly the coverage for a typical road can be represented) and prediction accuracy is application-dependent (Section 4.7.2). Hence, it is likely that there will be several extents required to span each road for each wireless network. Assuming that the extents for each network do not start and end at the same points along the road as any of the other networks, this model of the graph's complexity can be further developed. If there are q extents on a road, these will require the creation of $q-1$ virtual nodes and $q-1$ new edges (in addition to the single edge necessary to represent the road with one extent covering its whole) per road per network. This will result in an expansion to $|N'| = m|N| + m^2(q-1)|E|$ nodes and $|E'| = m|E| + m^2(q-1)|E|$ edges in the worst case.[7]

When handover nodes and edges are added, a handover node[8] is added to each edge in the graph, for each network. This is because for each node in the graph a handover node is created on each edge emanating from it, and this is done for each network as each may have a different handover time. One handover edge is then

[7] This occurs when, (other than the start and end nodes of the road) none of the extent boundaries for any of the m wireless networks are at the same location as one or more extent boundaries for other networks. In practice there is likely to be a small degree of co-location of extent boundaries.

[8] Note that only *one* rather than two handover nodes are added, because edges are always directed (i.e. one way). Hence despite the fact that each edge connects two nodes together, a handover node is only added on that edge by one of them.

added for each handover node added. This results in the graph with handovers, $G'' = (N'', E'')$ having:

$$|N''| = |N'| + (m-1)|E'| \tag{5.1}$$
$$|E''| = |E'| + (m-1)|E'| \tag{5.2}$$

Hence, the overall graph now has size

$$|N''| = m|N| + m(m^2q + m^2 + m - 1)|E| \tag{5.3}$$
$$|E''| = m^2(mq - m + 1)|E| \tag{5.4}$$

which has complexity $O(m^3)$ in terms of m, and $O(q)$ in terms of q.

5.4.2 Reducing Complexity Using Sparse Planes

Clearly, this graph expansion will be very large if m and q are not trivially small. To mitigate the expansion due to q (the average number of extents per road) the algorithm used to generate the extents could be adjusted, trading accuracy of RSS prediction for compactness as q is decreased. To counter the more significant effect of m, the number of wireless networks, it is important to note that many networks (particularly hotspot-oriented technologies such as WiFi) are only present over a very small subset of the overall graph G. For example, the WiFi network at the University of Cambridge Computer Laboratory is only accessible from the few roads that surround the building. A simplistic solution would be to create a plane that only included those edges that the wireless network was available on: a *sparse* plane. The issue with this approach is that it does not cater for occasions where it might be advantageous for a user to tolerate a brief disconnection but remain *notionally* connected to a particular network, thus avoiding the penalty of a handover to another network. Figure 5.6 illustrates the problem for a short journey where a user would be forced to handover twice. This would take a total of 9 seconds of disconnection time, compared to only 3 seconds[9] had they foregone connectivity solely from B to C. This would waste some of the available coverage of network 1 (a portion of the edge between nodes C and D).

5.4.3 Zero-Coverage Planes

To counter the issue depicted in Figure 5.6, in addition to creating the sparse planes a number of *complete* planes (i.e. ones that represent the underlying road topology exactly) are also created. A complete plane is added for each wireless network *technology* that exists in the graph. Note that this is instead of for each individual *network*, as there exist many networks of each technology; e.g. there are many different cellular network providers, but the majority use only GSM or UMTS.

[9] Plus the time cost of the re-connection to the network when reaching node C, described later.

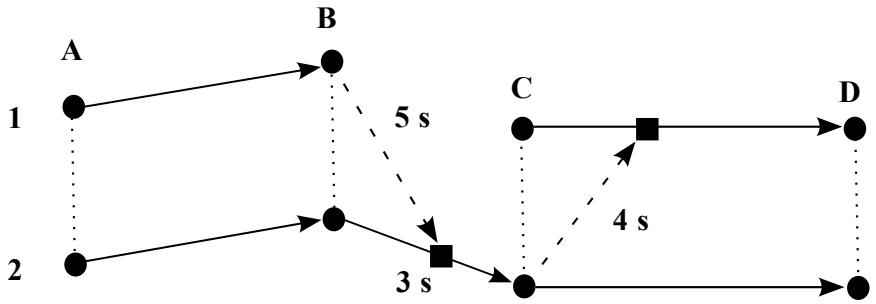


Figure 5.6: A problem with sparse planes. When travelling from node A to node D on network 1, a user is forced to perform a handover to network 2, and back again (total 9 seconds). This is of lower utility than tolerating a brief disconnection between nodes B and C whilst remaining on network 1 (3 seconds), plus a small amount of time to reconnect to the network at C.

The complete planes are special in that the weights of all their edges indicate the absence of wireless coverage, and hence they are termed *zero-coverage planes*. Evidently there will be geographical areas that a zero-coverage plane spans that do in fact have coverage of one or more wireless networks of the same technology type as the zero-coverage plane. This does not present a problem, as a routing algorithm will simply ignore the edges which are of lower utility (i.e. those in the zero-coverage plane) and prefer the corresponding edges in the sparse planes of non-zero network coverage.

Zero-coverage planes solve the problem shown in Figure 5.6 by permitting handovers from a wireless network of a particular technology to the zero-coverage plane for that technology in zero time. The edge between the source node and the node in the graph that the handover edge connects to is "vertical". Meanwhile, handovers *from* a zero-coverage plane have a time cost that is equal to the standard handover time from the technology of the zero-coverage plane to the technology of the destination plane. This includes an edge from the zero-coverage plane to any sparse planes of the same technology. Such intra-technology (horizontal) handovers have a time cost that is lower than that of vertical (inter-technology) handovers. This is because aspects such as the magnitude of the throughput, latency, and packet loss rate will be approximately the same on the source and target networks.

For completeness, handover edges are created from all planes to the zero-coverage planes. The handover time from a plane whose technology is not the same as that of the destination zero-coverage plane will be the usual time for a vertical handover between the two technologies, v, minus the horizontal handover time for the zero-coverage plane's technology, h. This implies that when moving from a network of technology 1 to the zero-coverage plane of technology 2, and thence to a real

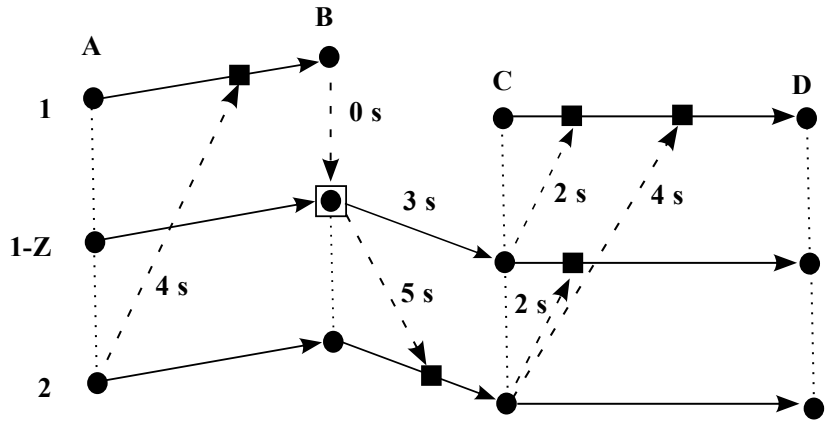


Figure 5.7: Using zero-coverage planes for horizontal handovers. Only a selection of handover nodes and edges are shown for simplicity; see key in Figure 5.4 for node meanings. Network 1 has a sparse plane, and has a zero-coverage plane of the same technology (1-Z), whilst network 2 is available throughout the path from A to D. The disconnection time when travelling from C to D is now only 2 seconds longer than the traversal time of the edge, totalling 5 seconds of disconnection rather than 9 seconds as in the original example.

network of technology 2, the total handover time is that which would be expected for a vertical handover from technology 1 to technology 2, i.e. $(v - h) + h = v$. A selection of the various possible handovers using zero-coverage planes is illustrated in Figure 5.7.

It is worth noting that this is representative of reality: it is never possible to connect and register with a wireless network without being in range of it. Connecting takes a finite period of time, which here is shortened using knowledge of the coverage area of the network, rather than waiting a little to ensure that it is not a fleeting (and hence useless) contact. Hence, the proposed model of requiring a horizontal handover time even when coming from a zero-coverage plane of an identical technology is realistic. This also explains why handover edges are *not* inserted that emanate from handover nodes (handover edges only ever *terminate* at handover nodes): a handover may only begin to take place where the destination wireless network's coverage is present.

Another point of note is that some networks, particularly cellular networks, are almost ubiquitous. This means that using sparse planes to represent each provider's network, and in addition providing a zero-coverage plane for that technology might well be a waste. Instead, for those small areas where there is no coverage, zero-coverage extents are included in the "sparse" plane, thus making it a full plane. This has no effect on the arguments presented above.

Term	Overview	Section	Figure
Junction Node	A node in the road graph where edges meet (e.g. crossroads, roundabouts).	5.2	5.2
Virtual Node	Represents the boundary between two extents.	5.3.1	5.4
Handover Node	The point of termination of a handover edge.	5.3.2	5.5
Complete Plane	Used for ubiquitous technologies, has edges for every road.	5.4.3	5.7
Sparse Plane	Used for hotspot technologies, covers only a few roads.	5.4.2	5.6
Zero-Coverage Plane	Complete plane, one per technology, representing no coverage.	5.4.3	5.7
Vertical Handover	Handover between technology types.	5.3.4	5.5
Horizontal Handover	Handover between different networks of the same technology, probably via a zero-coverage plane.	5.4.3	5.7

Table 5.2: A glossary of terms introduced concerning the proposed multi-planar graph structure, with their corresponding Section and Figure numbers.

5.4.4 Complexity of Sparse & Zero-Coverage Planes Approach

Expressing the complexity of the overall graph when zero-coverage planes are used requires several assumptions to be made. Previously it was assumed that m was the number of distinct wireless networks. It is now assumed that there are t different wireless technologies present, with t_u of these having (near-)ubiquitous coverage, and t_s having sparse coverage ($t = t_u + t_s$). It is further assumed that there are u instances of each type of ubiquitous network, and s instances of each sparse network (e.g. there are s distinct WiFi hotspots). Also, for sparse coverage networks, it is assumed that these will only cover a fraction f, $0 \leq f \leq 1$ of all edges in the graph. This results in each network covering $f|E|$ edges (though of course the

edges for any number of wireless networks may overlap). The number of extents required for each road (on average), q is assumed to be the same for all wireless networks, regardless of technology.[10]

Firstly, the complexity of the graph $G' = (N', E')$ when planes have been added for all wireless networks, as well as zero coverage planes, is calculated. For each network of a technology that is ubiquitous $(q-1)|E|$ nodes and $(q-1)|E|$ edges are generated to be added to every other plane (in addition to each plane already being a copy of G, and therefore having $|N|$ nodes and $|E|$ edges). For those networks that are sparse, in the worst case where the subgraph is a simple path[11] there will be $f|E| + 1$ nodes and $f|E|$ edges in the subgraph of G that are relevant to each sparse network. When the extents are included these will generate $(q-1)f|E|$ extra nodes and $(q-1)f|E|$ extra edges per network plane.

For each ubiquitous network's plane that takes into account all networks' extents, $G'_u = (N'_u, E'_u)$ we have:

$$|N'_u| = |N| + (q-1)|E|ut_u + (q-1)f|E|st_s \quad (5.5)$$

$$|E'_u| = |E| + (q-1)|E|ut_u + (q-1)f|E|st_s \quad (5.6)$$

For each sparse network's plane, $G'_s = (N'_s, E'_s)$ an assumption must also be made concerning the number of other sparse networks that each sparse network overlaps with. The number of sparse networks that can be overlapping each other is termed g. In the worst case this overlap is complete, i.e. for each sparse network there are $g-1$ others that have extents on all the roads covered by the network under consideration. For simplicity it is assumed that g is fixed over the entire graph, and that st_s is a multiple of g. Hence for each sparse network we have:

$$|N'_s| = (f|E|+1)|N| + (q-1)f|E|ut_u + (q-1)f|E|g \quad (5.7)$$

$$|E'_s| = f|E| + (q-1)f|E|ut_u + (q-1)f|E|g \quad (5.8)$$

Overall, assuming that zero-coverage planes are only added for sparse network technologies:

$$|N'| = ut_u|N'_u| + t_s|N'_u| + st_s|N'_s| \quad (5.9)$$

$$|E'| = ut_u|E'_u| + t_s|E'_u| + st_s|E'_s| \quad (5.10)$$

Handover nodes and edges must now be added to the graph. This is simple, since on each edge a handover node is added for each of the network technologies that

[10] In practice this is unlikely to be true: cellular network coverage varies more gradually than does WiFi, for example. For the sake of simplicity this point is elided.

[11] By not allowing cycles or trees the subgraphs covered by the sparse networks are forced to have the maximum possible number of nodes in them, because each node will only have two edges connected to it.

are present in the entire graph, i.e. $t_u + t_s$ nodes are added to each edge[12]. Note that this is per technology rather than per individual network, as handovers from all networks of a particular technology to another technology are assumed to have the same costs. Edges must then be constructed from all existing nodes in the graph to the handover nodes that correspond to them. Hence,

$$|N''| = (t_u + t_s)|E'| - t_s|E'_u| \tag{5.11}$$

$$|E''| = (ut_u + t_s - 1)|N'| + (ut_u + t_s)st_s|E'_s| + st_s|E'_s|(g-1) + |N'_s|st_s \tag{5.12}$$

The first term in the expression for $|E''|$ represents the connections *from all* nodes to the ubiquitous and zero-coverage planes. One is subtracted from the number of planes, as nodes in ubiquitous planes do not have handover connections to their own planes. The second term represents the connections *to* each edge in the sparse planes from all ubiquitous and zero-coverage planes. The third term accounts for the interconnections between sparse planes, which is the total number of nodes in the sparse planes multiplied by the number of overlapping sparse networks, $g - 1$. However, account must also be taken that the first term included nodes in the sparse planes, which are (evidently) not located on a ubiquitous or zero-coverage plane, and hence would *not* be being connected to their own plane by the first term. The final term is added to account for this.

Combining all of these expressions together the most significant terms in the equations for $|N''|$ and $|E''|$ can be determined, i.e. those with the highest powers of u, t_u, s and t_s:

$$|N''| \approxeq ((ut_u + fst_s)(ut_u + t_s)(t_u + t_s) + fst_s(ut_ut_s - t_s + ut_u))(q-1)|E| \tag{5.13}$$

$$|E''| \approxeq ((ut_u + fst_s)(ut_u + t_s)(ut_u + t_s) + fst_s(u^2t_u^2 + ut_ut_s)(2))(q-1)|E| \tag{5.14}$$

5.4.5 Comparison of Both Approaches

These expressions should be compared to those derived earlier where no zero-coverage planes were used, and where complexity was $O(m^3)$. Given than $m = ut_u + st_s$, then m^3 will have most significant terms of $u^3{t_u}^3$ and $s^3{t_s}^3$. In a graph that does utilise zero-coverage planes we expect no change in complexity for ubiquitous network technologies (as these must have full planes in the graph), and hence the first term in the equation for $|E''|$ above also involves $u^3{t_u}^3$. Interestingly the expression for $|N''|$ has a slightly smaller term than would be expected, with $u^2t_u^3$.

[12]Except on zero-coverage edges, where one less handover node is added, as all handovers from networks of the same technology as that of the zero-coverage plane incur zero time penalty, and hence end at an already existing virtual node.

More importantly, for sparse networks we achieve a reduction by a factor of s^2/f. This is particularly significant since s (the number of networks of each sparse technology) is likely to be much greater than t_s (the number of sparse technologies), e.g. there are a huge number of private WiFi networks.

As part of the evaluation of the overall system (Section 6.5) statistics were collected concerning the multi-planar graphs the algorithm constructed. These are shown in Table 5.3, with the $|N''|$ and $|E''|$ columns indicating the actual size of the multi-planar graph generated from an original (road topology) graph containing $|N|$ nodes and $|E|$ edges. The test cases used one ubiquitous network (a UMTS cellular provider), and a large number of 802.11b/g WiFi hotspots, i.e. $t_s = t_u = 1 = u$, s being variable according to the trace, and the total number of networks being $m = (1)(1) + s(1) = 1 + s$. The figures for $f|E|$ were generated by calculating the mean number of edges per (sparse) wireless network in each trace, then multiplying this by the number of sparse wireless networks, s. Therefore, this figure is a worst case scenario that does not consider any overlap between networks[13], though it could be divided by a suitable value of g to give one. The figures show how utilising the concept of sparse planes significantly reduces the size of the overall graph as compared to the approach of generating one plane per distinct network, whose size would be proportional to $m^3|E|$, as given in the final column. That this approach reduces the size of the graph is not a surprise, but the magnitude of the reduction is crucial in justifying how the proposed scheme could be made to work on low-resourced devices. Evidently, however, the ratio of $|N''|$ to $|N|$ is still very large, and hence further reductions are required, likely to be achieved by decreasing q, the mean number of extents per road. In Chapter 4, five different extent generation algorithms were evaluated in terms of accuracy and compactness (the greater the mean length of all the extents generated by an algorithm, the more compact it was deemed to be). In Chapter 6, the algorithm previously found to be optimal for both these characteristics is used, but it is important to note that it is likely that many applications would tolerate a lower accuracy, and hence a lower value of q.

A further optimisation (not yet implemented), removes a large number of handover edges by making more use of the zero-coverage planes. Because every plane has zero cost handover connections to its corresponding zero-coverage plane, the condition that any other type of handover must be *via* the zero-coverage plane can now be imposed. This then means that each zero-coverage plane would have handover edges to all other planes (as currently), but that all non-zero-coverage planes would *only* have handover connections to their corresponding zero-coverage plane (and no others). Evidently such an approach would significantly decrease the complexity of the resulting graph.

[13] Because this is the worst case, where there *was* network overlap in some of the test runs $f|E| > |E|$ in Table 5.3.

#	$\|N\|$	$\|E\|$	$\|N''\|$	$\|E''\|$	$f\|E\|$	s	$m^3\|E\|$
1	24	23	1113	7776	5	5	4.97e+03
2	80	79	2603	21159	25	13	2.17e+05
3	85	84	3963	30999	20	15	3.44e+05
4	56	55	1449	9795	15	10	7.32e+04
5	65	64	1691	11577	17	11	1.11e+05
6	66	65	3301	27171	18	15	2.66e+05
7	59	58	3205	26460	19	14	1.96e+05
8	150	149	4189	33309	29	22	1.81e+06
9	33	32	547	3492	6	5	6.91e+03
10	49	48	1661	11709	19	11	8.29e+04
11	75	74	1948	13476	22	14	2.50e+05
12	73	72	2125	16095	19	11	1.24e+05
13	33	32	381	2475	4	3	2.05e+03
14	48	47	563	3351	2	2	1.27e+03
15	69	68	4687	37887	37	18	4.66e+05
16	76	75	2577	20928	25	13	2.06e+05
17	80	79	5448	44613	48	23	1.09e+06
18	59	58	3912	33363	43	21	6.18e+05
19	75	74	3979	33546	43	19	5.92e+05
20	68	67	4347	36474	47	25	1.18e+06
21	13	12	910	7056	4	3	7.68e+02
22	100	99	3197	26067	13	10	1.32e+05
23	76	75	4654	38643	36	17	4.37e+05
24	55	54	2101	16494	25	12	1.19e+05
25	21	20	792	4971	2	2	5.40e+02
W_1	5731	6113	76430	633255	170	90	4.61e+09
W_2	5731	6113	41637	290883	178	90	4.60e+09

Table 5.3: Statistics of graph complexities for each trace used in the evaluation in Section 6.5 using density-dependent dominant point detection, and also the whole city using a coverage map generated by density-dependent dominant point detection (W_1), and another using Nearest Neighbour Interpolation (W_2). The latter is less accurate but covers a greater number of roads.

By decreasing the complexity of the graph, routing algorithms will run with lower time complexities, as well as the graph itself requiring less storage resources. However, it is important to note that the generation of the graph does *not* need to take place onboard a vehicle. Instead, it can be generated elsewhere, and downloaded onto the vehicle for use. Hence, finding a route that is optimised not only for distance or time but also for wireless coverage would not involve all the processing

to create the graph that described in this Chapter, and hence the processing power of, say, a portable satellite navigation device would be likely to be sufficient. It is important to note that the proposed method, despite its usage of zero-coverage planes, does increase the number of nodes and edges by significant factors. For an entire city (W_1) for which a relatively small proportion of roads are covered, the number of nodes increases by a factor of 13.3 and the number of edges by 103.6. With a more comprehensive coverage map this might increase to a factor of up to 500, as exhibited in each test trace row of Table 5.3. Clearly, the current generation of portable navigation devices are not designed for such an increase in graph size. However, such graph sizes are more than tractable on a standard workstation, and processing power onboard vehicles is growing rapidly as more general purpose operating systems are deployed. Hence, it is reasonable to expect that as navigation devices gain capabilities that are already in the pipeline, such as 3-D or photorealistic views of cities, storing and processing the graphs generated by the proposed scheme will not present a particular challenge.

5.5 Chapter Summary

Leading on from the coverage mapping algorithms described in Chapter 4, this Chapter has shown how such coverage maps can be used. The value of QoS-aware routing was illustrated, along with its possible use cases. Next, an explanation was given as to why the intuitive approach of simply augmenting a graph with RSS values does not work, before moving on to consider in detail how coverage extents can be used to construct a multi-planar graph. The complexity of the resulting network was analysed, and sparse and zero-coverage planes were proposed as significant complexity-reduction techniques. Chapter 6 will proceed to use the multi-planar graph for QoS-aware routing.

6

QoS-Aware Multi-Criteria Routing

THIS Chapter presents how the multi-planar graph that represents the coverage of multiple networks described in Chapter 5 can be used. With an appropriate routing metric, it is possible to calculate an efficient sequence of handovers that should be carried out on a journey between two locations. However, constructing such a routing metric requires a knowledge of the theory of maximisable routing metrics, as we wish to maximise network QoS but minimise distance travelled. Such multi-criteria routing permits routes to be found that have higher mean throughputs compared to the shortest geographical paths, whilst not being substantially longer. In this Chapter, a family of routing metrics is proposed that aims to maximise mean throughput over a journey, whilst minimising handovers, and ensuring that the distance to be travelled is also taken into account. Each proactive metric is compared on real traces to a reactive handover algorithm, to show how much benefit a proactive algorithm provides. Then, each proactive metric is used to find the best QoS route between random pairs of locations. This is compared to the QoS achievable over the shortest path route between the two points, to ascertain whether route selection (as well as network selection) is successfully performed by the proactive algorithms.

6.1 Properties of Routing Metrics

Traditional shortest-path routing for vehicles involves representing the road network as a directed graph $G = (N, E)$, where N is the set of road junctions, and E the set of roads connecting them; a function $d : E \to D \subseteq \mathbb{R}_0^+$ that provides the geographical length (distance) of each edge; and a function $w : E \to W \subseteq \mathbb{R}_0^+$ that provides a weight for each edge. In the simplest case $w(e) = d(e)$, where the weights assigned to the edges are simply their lengths. The goal of shortest-path routing is to find a simple path between two given nodes which comprises a set of edges $R \subseteq E$ (a route), taken from the set of all possible routes $\mathscr{R}$ between the two nodes, such that $\sum_{e \in R} w(e)$ is minimised.

6.1.1 Requirements for Globally Minimisable Routing Metrics

Two well-known algorithms that exist for solving the shortest-path problem are Dijkstra's [62] and that due to Bellman [16], Ford [122], and Moore. Both are able to solve the shortest-path problem as described above. When defining the weighting function w in order that a particular attribute (such as total length of route) is minimised globally over the route, three conditions must be satisfied by w. Here, it is assumed that the attribute to be minimised is mapped onto edges by a function $x : E \rightarrow X \subseteq \mathbb{R}_0^+$, where X is the set of values that the attribute may take:

- **Monotonicity:** $\forall_{e_1,e_2 \in E} x(e_1) > x(e_2) \Rightarrow w(e_1) > w(e_2)$
- **Maximality:** $\forall_{e \in E} w(e) \geq 0$
- **Homomorphism:** $\forall_{e_1,e_2,e_3 \in E} x(e_3) = x(e_1) + x(e_2) \Rightarrow w(e_3) = w(e_1) + w(e_2)$

The first condition ensures that the metric increases as the quantity to be minimised increases, whilst the second avoids the existence of any negative metrics (see below). The right-hand side of the third condition can be re-expressed as $w(e_1 \oplus e_2) = w(e_1) + w(e_2)$, where $\oplus$ signifies composing edges e_1 and e_2 into a path, and returning a single edge that is equivalent (in all its attributes) to that path. This formulation states that w is a homomorphism over the $+$ operator, and is necessary in order to ensure that the *sum* of the attribute (x) over the path is the global minimum possible. A simple example where this condition does not hold, resulting in a route that is not the shortest path is illustrated in Figure 6.1. These three conditions are similar to those given by Sobrinho, who proved that for a routing metric to converge to the globally shortest path, it is both necessary and sufficient that the metric function be both monotonic and isotonic [212]. His definition of isotonicity can be phrased as

$$\forall_{e_1,e_2,e_3 \in E} w(e_1) \leq w(e_2) \Rightarrow w(e_1 \oplus e_3) \leq w(e_2 \oplus e_3) \tag{6.1}$$

The requirement for w to be homomorphic over $+$ subsumes this condition, as it can be seen that

$$\begin{aligned} \forall_{e_1,e_2 \in E} \quad & w(e_1) \leq w(e_2) \\ \Rightarrow \quad & \forall_{e_3 \in E} w(e_1) + w(e_3) \leq w(e_2) + w(e_3) \\ \Rightarrow \quad & w(e_1 \oplus e_3) \leq w(e_2 \oplus e_3) \end{aligned}$$

and hence isotonicity is also satisfied by it.

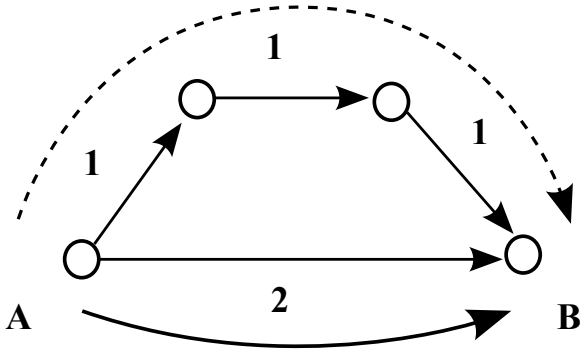


Figure 6.1: An illustration of the need for a routing metric to be homomorphic over $+$. The figures on the edges represent their lengths, $d(e)$. If $w(e) = d(e)$ then the shortest path from A to B would be calculated to be as indicated by the solid curved arrow. If $w(e) = d(e)^2$ (not homomorphic over $+$) then the path would be as indicated by the dashed arrow. The latter would clearly be incorrect were our goal to be to minimise the sum of edge lengths.

6.1.2 Composition of Edge Properties

An important point concerning composition is how it takes on a different meaning depending on the property being considered. For distance-based routing, if $e_3 = e_1 \oplus e_2$ then $d(e_3) = d(e_1) + d(e_2)$, but other attributes such as throughput, $b(\,\cdot\,)$, do *not* compose additively (clearly it does not hold that $b(e_3) = b(e_1) + b(e_2)$). This fact will be important later on, when edge composition is defined to include throughputs and data transfers.

6.1.3 Complexities of QoS-aware Routing

The scenario introduced in this work is more complex than the simple shortest-path problem for two reasons:

- There are three quantities associated with each edge that the routing algorithm must take into account, namely edge length, throughput available over that edge, and whether it is a handover edge.
- The aim in this work is to minimise the distance (or analogously, time) to traverse a route between two given points, but also to maximise what network QoS experienced along it.

The first of these points seems relatively simple to solve. To formalise the problem, a third function $b : E \to B \subseteq \mathbb{R}_0^+$ is defined that assigns a throughput to each edge,

and a fourth function to assign traversal times to each edge, which for simplicity is taken as proportional to their length, i.e. $t : E \rightarrow T \subseteq \mathbb{R}_0^+$ where $t(e) = d(e)/v(e)$ and $v : E \rightarrow V \subseteq \mathbb{R}^+$ gives the velocity on each edge. A final function $h : E \rightarrow H \subseteq \mathbb{Z}_0^+$ indicates the number of handovers taking place over the edge (zero or one, but we allow h to take values of greater than one for when it is applied to a path, as outlined below). A function $f : (B \times T \times H) \rightarrow W$ must now be found to allow the edge weights to be set to $w(e) = f(b(e), t(e), h(e))$.

Turning to the second point, initially, the need to minimise route traversal time will be ignored, and instead the focus will solely be on how to maximise a quantity such as the amount of data that could be transferred over a route. For each edge, the amount of data that can be transferred is $b(e)t(e)$. Hence, a naïve method to attempt to achieve a maximisation of data transferred would be define $w(e) = \frac{1}{b(e)t(e)}$, i.e. invert the metric. The assumption is made that if

$$R_{\min} = \{R \in \mathscr{R} | \min(\sum_{e \in R} \frac{1}{w(e)})\} \tag{6.2}$$

$$R_{\max} = \{R \in \mathscr{R} | \max(\sum_{e \in R} w(e))\} \tag{6.3}$$

then $R_{\min} = R_{\max}$. Unfortunately this does not hold true, as for two numbers $a, b \in \mathbb{R}^+$,

$$\frac{1}{a} + \frac{1}{b} > \frac{1}{a+b} \tag{6.4}$$

and hence for two routes having equal values of $\sum_{e \in R} b(e)t(e)$ it would *not* be the case that they had equal values of $\sum_{e \in R} w(e)$. This would then mean that the routing algorithm would not be guaranteed to find the route with the maximal amount of data transferred: the sum of a valid routing metric must *decrease* as more data is transferred. Instead, with this method, each edge that is added, in fact *adds* to the sum, albeit only a small quantity.

6.1.4 Requirements for Globally Maximisable Routing Metrics

Gouda and Schneider [91] outline two properties that must hold for a routing metric to be *maximisable*: boundedness and monotonicity. The boundedness property states that given a route R, having a total metric $m = \sum_{e \in R} w(e)$, if an edge is added to produce $R' = R \cup \{g\}$ that has metric $n = \sum_{e \in R'} w(e)$ then $n \leq m$. This is necessary since it is wished that "longer" (or in this case, higher data transfer) routes to be preferred, and shortest-path algorithms prefer paths with *lower* total weights. It is therefore analogous to Sobrinho's monotonicity property for path *minimisation*, but with a change of sign in the inequality. It is now evident that the reason the $\frac{1}{b(e)t(e)}$ metric described in the previous Section fails is because it violates this boundedness property. Gouda and Schneider's second property, monotonicity, is (confusingly) the same as Sobrinho's isotonicity condition.

A plausible course of action to achieve a maximisable routing metric that satisfies Gouda and Schneider's conditions would be to modify w to yield negative numbers, i.e. $w : E \rightarrow \mathbb{R}_0^-$. Clearly this can satisfy their boundedness and monotonicity properties. Unfortunately, if the graph $G = (N, E)$ represents the road network, in the vast majority of cases cycles will be present, i.e. there will exist routes consisting of non-zero numbers of edges that each begin and end at a single node. If all weights are negative, then evidently the sum of the weights of the edges in a cycle will also be negative. This being the case, traditional shortest-path algorithms will not arrive at a solution, as any route that incorporates a cycle can be made to have a lower sum of weights by traversing the cycle once more than the route currently specifies. Shortest-path algorithms do exist that are able to detect the presence of negative cycles [34], but evidently such a capability is of limited interest if once a negative cycle is detected a route cannot be found.

6.1.5 Inefficiency of Solving the Maximisation Problem

Given the above, global *maximisation* of a particular metric does not appear to be efficiently soluble using shortest-path algorithms. In actual fact, the Longest Path Problem is the general case of the Travelling Salesman Problem, which is known to be $\mathcal{NP}$-hard [100]. Algorithms have been devised to generate such longest paths for restricted cases, such as the resulting longest paths having a cardinality of less than a specified limit [200], or on particular classes of graph such as weighted trees [226]. However, such algorithms are of no use in the case of a real-life road network, particularly since the aim here is to provide a solution that plausibly could be implemented on a portable navigation device.

Therefore, the aim of this work is *not* to provide the *optimal* routes as regards maximising (e.g.) data transferred. Instead, recognising the limitations of using shortest-path algorithms, but doing so for processing efficiency, metrics are developed and evaluate that will *improve* network QoS as compared to the existing approach of simply constraining the geographical route to the shortest path, and then attempting to ascertain the sequence of networks that should be used over that route.

Concretely, the aim is to locate *efficient* solutions to a problem where multiple routes may be equally good overall, but nonetheless have different characteristics. This is the area of multiobjective routing.

6.2 Overview of Multiobjective Routing

Multiobjective programming is well established field [36]. Most problems are expressed as a goal that involves the maximisation of a particular function, and several constraints. All of these are linear. Although there are many problems that can be expressed with non-linear constraints, there are no known algorithms to efficiently find the optimal solutions.

In terms of routing, many authors have considered how people trade-off time and cost. For example, a driver may take a route that involves a toll road, which is short in time but economically costly. If the toll is too high, she will take a different (longer) route that costs less. Thus, the driver has a total cost function that consists of a weighted sum of that due to lost time, and that expended on tolls. The aim is to minimise that weighted sum. Clearly there may exist multiple solutions that have equally low total costs: these are termed *non-dominated* or *efficient* solutions. The collection of such solutions is known as the *Pareto set*.

Concretely, if for each road e_i there is a time cost $t(e_i)$ and an economic cost $c(e_i)$, then the total cost for that road is of the form

$$u(e_i) = at(e_i) + bc(e_i)$$

where a and b are user defined constants depending on the trade-off of time and economic cost. A path P_j composed of such roads will have cost $U(P_j)$:

$$U(P_j) = \sum_{e_i \in P_j} u(e_i)$$

If the goal is to minimise the total cost, then a non-dominated solution is a path P^* for which the following holds:

$$\forall_{P_j} U(P^*) \geq U(P_j) \Rightarrow U(P^*) = U(P_j)$$

These solutions make up the Pareto set.

6.2.1 Pareto Optimality Versus Lexicographical Ordering

An important point concerning Pareto optimality is that it is distinct from lexicographical ordering. If each road is assigned r different properties (of which time and economic cost might be two), a routing algorithm must consider an r-vector per road. Here, two example roads $\mathscr{X}$ and $\mathscr{Y}$, with r-vectors $x = (x_1, \ldots, x_r)$ and $y = (y_1, \ldots, y_r)$ are considered. Under a lexicographical ordering that uses a relation $\prec$, $x \prec y$ *iff*:

$$\begin{aligned} x_1 &< y_1 \vee \\ x_1 &= y_1 \wedge x_2 < y_2 \vee \\ &\vdots \\ x_1 &= y_1 \wedge \ldots \wedge x_{r-1} = y_{r-1} \wedge x_r < y_r \end{aligned}$$

This then means that two edges may only be considered equal if $\forall_{k\in[1..r]} x_k = y_k$. In contrast, solutions in the Pareto set may have r-vectors that have no coincident elements, but that do not dominate each other. If the set of all possible r-vectors is D, then a vector $\mathbf{d} \in D$ is Pareto optimal iff there exists no other $\mathbf{d}' \in D$ where $\forall_{j\in[1..r]} \mathbf{d'_j} \leq \mathbf{d_j}$ and for at least one j, $\mathbf{d'_j} < \mathbf{d_j}$. Therefore, if an efficient solution is found by shortest path routing considering *each* of the r properties in isolation, then each of these r solutions will be in the Pareto set for the overall problem.

6.2.2 Extreme Non-dominated Solutions

Which of the solutions in the Pareto set are in fact the *optimal* solutions to the problem depends on the definition of the utility function. If the utility function is quasiconvex then the optimal, or *extreme non-dominated*, solutions will be contained within the set of points that make up the convex hull of the Pareto set ([15] (Theorem 3.5.3)). This is illustrated in Figure 6.2. By requiring the utility function to be quasiconvex, it is ensured that if each of the r properties are monotone decreasing, the utility function must be a linear combination of the r properties. Hence:

$$\begin{aligned} &\text{If } J = \{j \in \{1..r\} | x_j < y_j\} \text{ then} \\ &\qquad u(\mathscr{X}) < u(\mathscr{Y}) \Rightarrow J \neq \emptyset \end{aligned}$$

Once the set of extreme non-dominated solutions is found, search algorithms must be used to find the actual optimal solutions to the problem. Therefore, in contrast to lexicographical ordering, this paradigm allows trade-offs to be expressed (e.g. how far a driver is prepared to travel to avoid a particular toll), rather than only strict preference orders.

6.2.3 Generating the Pareto Set

A range of algorithms have been proposed to find the Pareto set in multicriteria problems. A few of these are overviewed here, to give an illustration of the wide variety.

Dial [60] gave an algorithm that calculated n separate routes that each minimised $U(P_j)$ for a different pair of values (a, b). The algorithm first calculated what the optimal route would be with $(a = 0, b = 1)$, then the reverse with $(a = 1, b = 0)$, before recursively deriving other routes that had parameter values between these. Each would have a different overall cost, but such an approach allowed a user to select among the results without needing to specify their values of a and b *a priori*.

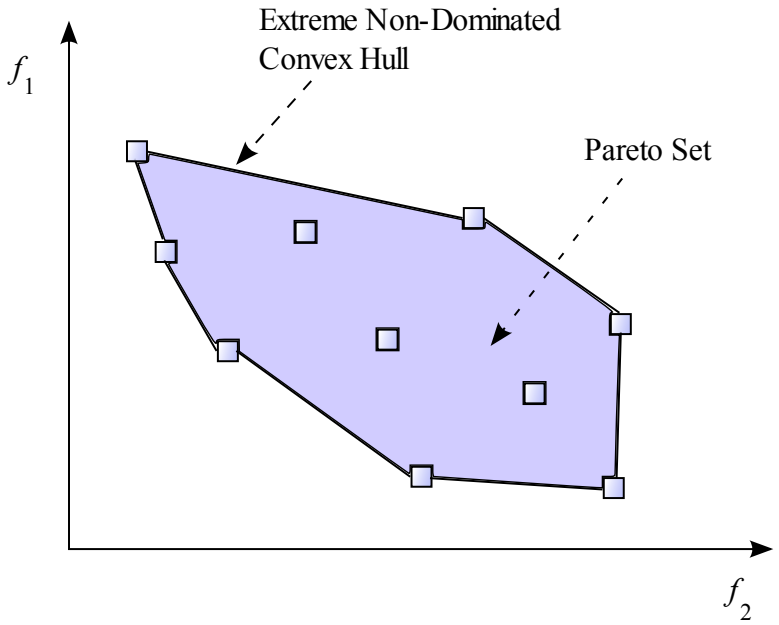


Figure 6.2: The non-dominated solutions (Pareto set) for a problem with two properties (f_1 and f_2). The convex hull of such solutions contains those which are extreme non-dominated. Several of these may be optimal solutions that have equal values of utility.

In a similar vein, Henig [104] set out to generate the Pareto set of solutions by finding the *extreme path* for each value pair of a and b (where in this case $a = 1 - b$). Having generated this set of extreme paths, several search methods were proposed to find the *optimal* solution from among them, depending on how $U(P)$ was defined.

Warburton [237] pointed out that by using such weighted linear combinations of objective functions and considering each in turn as a single-objective, shortest path problem, Henig (and others) risked missing large segments of the solution space. Instead, Warburton proposed that the routing algorithm keep track of all the non-dominated paths found thus far to the node currently under consideration. When the next edge was added to each of the paths, those that were non-dominating would be deleted. However, such an approach has very high space complexity. Therefore, algorithms were also proposed to reduce it to more manageable levels.

Another related idea is the use of simulated annealing to produce more optimal solutions from a generating set [50]. To the author's knowledge, this has not yet been applied to routing, but appears a relevant technique.

6.2.4 Routing with Conflicting Criteria

Multicriteria routing as applied to transportation networks is a well researched field, as Current *et al.*'s surveys on the subject attest to [49, 48]. The majority of relevant work has concerned objectives that do not conflict, the goal being, e.g. to minimise the sum of each of the criteria. However, some work has examined the commonly occurring (but $\mathcal{NP}$-hard) problem domain where criteria conflict. Outside the field of routing, a well known example of conflicting criteria is the Knapsack problem, where space in a container is limited, but the most useful objects to be placed in that container are not necessarily the smallest.

Perhaps the most relevant papers are those concerning the orienteering problem (OP) [225]. This is similar to the travelling salesman problem, but there is a reward for visiting each node, as well as a distance penalty in travelling to it. The objective is to maximise reward and minimise penalty. The constraint method is one approach proposed for solving this Multiobjective Vending Problem, where a bound is placed on the total penalty, and a route found which has the maximum reward [132]. The penalty threshold is then increased and the process repeated. In this way the best trade-off route can be found. This method was compared to three other algorithms, finding it worked best overall in the three scenarios that were tested.

Park and Koelling [184] examined the problem of how to route a vehicle to visit various depots in the minimum distance/time, but whilst attempting to ensure that those depots that had to be urgently visited were done so quickly. In addition, some depots were potentially dependent in some way on others. Hence, the vehicle had to ensure that such dependencies were met as much as possible by visiting such subsets of depots in the correct order. These criteria evidently sometimes conflicted with each other. The authors formulated various constraints on the route (such as a maximum time length before the product being carried might deteriorate), then formed a linear combination of them. This utility function was too complex to find the optimal solution by direct goal programming. Instead the authors proposed an iterative procedure, where the next step in the route was found by a linear goal programming model, and was contingent on the route selected thus far. The key point concerning this approach is that it is *not* guaranteed to generate optimal solutions, but nonetheless achieves results that satisfy the original goals.

6.3 A Family of QoS-aware Routing Metrics

Returning to the two original problems given in Section 6.1.3, a function f must be found that combines the quantities that are to influence the edge weights, and in addition what is meant by the intention to minimise route traversal time whilst maximising (as far as is possible) network QoS must be defined. The quantity of

data transferred over a route is an important aspect of network QoS, but interestingly here there is the problem that whilst the aim is to minimise route length (and hence traversal time), it is also to maximise it to gain a greater data transfer. In contrast, quantities such as the period of time spent disconnected, or the number of handovers, are both ones that should be minimised, and hence would be easy to incorporate into a shortest path metric.

New conditions that should apply to functions used as metrics must also be formulated. When two edges e_1 and e_2 are composed to form a path p (which will be treated as a single edge with the combined properties of e_1 and e_2), the properties of the path are defined to be as follows:

$$\begin{aligned} p &= e_1 \oplus e_2 \Rightarrow \\ w(p) &= w(e_1) + w(e_2) \\ b(p) &= \frac{b(e_1)t(e_1) + b(e_2)t(e_2)}{t(e_1) + t(e_2)} \\ t(p) &= t(e_1) + t(e_2) \\ h(p) &= h(e_1) + h(e_2) \end{aligned} \tag{6.5}$$

The expression for $b(p)$ ensures that the data that would be transferred over the path p is equal to the total that would be transferred if the two edges e_1 and e_2 were traversed, i.e. $b(p)t(p) = b(e_1)t(e_1) + b(e_2)t(e_2)$.

6.3.1 Criteria for a QoS-aware Metric

Given this definition of how edges with multiple metrics can be composed, the new conditions on the weighting function can be defined as:

- **Maximality:** $\forall_{e \in E} f(b(e), t(e), h(e)) \geq 0$
- **Homomorphism:** $\forall_{e_1,e_2 \in E} f(b(e_1 \oplus e_2), t(e_1 \oplus e_2), h(e_1 \oplus e_2)) = f(b(e_1), t(e_1), h(e_1)) + f(b(e_2), t(e_2), h(e_2))$

For the monotonicity condition, the aim is not solely to maximise the data transferred, but instead to do so whilst taking into account the length of time required for such a transfer. Because of this, when comparing two edges, both their ratios of amounts of data transferred and traversal times must be examined. The conditions for monotonicity are therefore as follows (note that all of these expressions apply to $\forall_{e_1,e_2 \in E}$):

- **If the amounts of data transferred are equal:**

$$\begin{aligned} b(e_1)t(e_1) = b(e_2)t(e_2) \quad &\wedge \quad t(e_1) = t(e_2) \\ \Rightarrow f(b(e_1), t(e_1), h(e_1)) \quad &= \quad f(b(e_2), t(e_2), h(e_2)) \end{aligned} \tag{6.6}$$

$$\begin{aligned} b(e_1)t(e_1) = b(e_2)t(e_2) \quad &\wedge \quad t(e_1) < t(e_2) \\ \Rightarrow f(b(e_1), t(e_1), h(e_1)) \quad &< \quad f(b(e_2), t(e_2), h(e_2)) \end{aligned} \tag{6.7}$$

- **Unequal amounts of data transferred, but obvious choice (prefer more data and less time):**

$$\begin{aligned} b(e_1)t(e_1) > b(e_2)t(e_2) \quad &\wedge \quad t(e_1) \leq t(e_2) \\ &\Rightarrow \quad b(e_1) > b(e_2) \\ \Rightarrow f(b(e_1), t(e_1), h(e_1)) \quad &< \quad f(b(e_2), t(e_2), h(e_2)) \end{aligned} \tag{6.8}$$

- **Unequal amounts of data transferred, but unpredictable choice:**

$$\begin{aligned} b(e_1)t(e_1) > b(e_2)t(e_2) \quad &\wedge \quad t(e_1) > t(e_2) \\ \wedge\, (b(e_1) > b(e_2) \quad &\vee \quad b(e_1) < b(e_2)) \\ \text{Let } n = \frac{b(e_1)t(e_1)}{b(e_2)t(e_2)} \quad &> \quad 1 \\ \text{Let } m = \frac{t(e_1)}{t(e_2)} & \\ \wedge\, m \quad &> \quad \alpha n \\ \Rightarrow f(b(e_1), t(e_1), h(e_1)) \quad &> \quad f(b(e_2), t(e_2), h(e_2)) \end{aligned} \tag{6.9}$$

The comparison of m and n is made using a user-defined parameter $\alpha \in \mathbb{R}, \alpha \neq 0$. Given two edges e_1, e_2, this allows us to express how much greater the ratio between e_1 and e_2's traversal times has to be compared to the ratio of their data transfers, before the weighting of e_1 is made larger than that of e_2. This expresses the case where it is not worth the time to travel down e_1, even though it has a higher amount of data transferred.

These conditions express (as far as is possible) the use cases given in Section 5.1.2. Note that a simple lexicographical ordering, where either the times or throughputs of two edges are compared, and only if this characteristic is equal are they then compared using another characteristic, would not work. This is because it would optimise only whichever characteristic was first compared, and not attempt to find routes where data *transfer size* is best. The conditions also express the need to not only consider the data transfer potential of an edge, as users will not want to solely maximise this quantity whilst not caring about the total traversal time. By trading off time and data transferred the hope is to obtain routes that are efficient in terms of traversal time, whilst transferring as much data as possible. The final use case of transferring a certain amount of data by the time the user arrives at their destination is not satisfied by these conditions: this is considered further in Section 6.7.

The approach in this dissertation is to attempt to satisfy these conditions as far as is possible, but knowing that no formulation of f will successfully satisfy all of them, as otherwise it would be possible to solve a maximisation problem in polynomial time. Hence, any formulation is not guaranteed to arrive at routes that are globally optimal.

6.3.2 General Form

The proposed formulation of f is

$$f(b(e), t(e), h(e)) = (b_{\max} - b(e))^n t(e) + t(e) + h(e) \tag{6.10}$$

where $b_{\max}$ is the maximum throughput that is possible over all the wireless technologies that are known about in the coverage map, and $n > 0$. This formulation encapsulates the need for an edge's weight to decrease as its throughput increases (as the difference between $b_{\max}$ and $b(e)$ decreases), but also makes the effect of this difference depend on the traversal time of the edge. As traversal times increase, bigger throughput differences are multiplied by a larger number, and thus the weight becomes much larger. This is representative of how users do not wish to travel along very long edges with low throughputs, but do not mind travelling along short edges with low throughputs if they lead to other edges that have high throughputs. The value of n allows the effect of the throughput difference to be changed, with smaller throughput differences having more significance as n is increased. The final $t(e)$ term is added in order that the edge will have a non-zero metric even if $b(e) = b_{\max}$: if all edges in the graph had such throughputs, routes would be found on the basis of the values of $t(e)$ alone, which appears to be a sensible course of action. Finally, the $h(e)$ term is added in order to allow the possibility of penalising a route that involves a large number of handovers. A multiplicative factor such as $\beta h(e)$ could also be used to change how much of an effect handovers have on the weighting. Using such a β would not affect the validity (or otherwise) of the properties of this formulation of f that are outlined below. Similarly, a minimum target throughput can be expressed by changing the value of $b_{\max}$ to instead be the target throughput. Any values of $b(e)$ greater than the target set to be equal to the target when used by the weighting function (in order that a negative metric is never obtained).

6.3.3 Satisfaction of Criteria by the General Form

It must now be ascertained what properties this formulation satisfies. For these proofs, n is set to 1, as this ensures that the majority of the conditions are satisfied. In Section 6.5 values of $n > 1$ are evaluated to see if (despite not satisfying the conditions), these yield better results.

Interestingly, allowing n to vary in order that the difference between the edge throughput $b(e)$ and the maximum possible throughput $b_{\max}$ might have a greater significance in the overall weighting, can be done another way. If $b_{\max}$ is allowed to be set as high as the user wishes, this then means that the trade-off of $\frac{t(e_1)}{t(e_2)} > \frac{b_{\max}-b(e_2)+1}{b_{\max}-b(e_1)+1}$ given above will be satisfied more for a greater range of values as $b_{\max}$ is increased. Thus, whilst in the evaluation n is varied, it should be noted that changing the value of $b_{\max}$ would preserve the conditions that were proven to hold for the metric when $n = 1$.

Maximality: given that $\forall_{b \in B} b_{\max} \geq b$, $\forall_{t \in T} t \geq 0$, and $\forall_{h \in H} h \geq 0$, clearly $\forall_{e \in E} f(b(e), t(e), h(e)) \geq 0$, and hence maximality is satisfied.

Homomorphism: given two edges $e_1, e_2 \in E$, these can be composed together to produce an equivalent edge $e_3 = e_1 \oplus e_2$. The weighting for this edge would be $f(b(e_3), t(e_3), h(e_3))$:

$$\begin{aligned}
& = (b_{\max} - b(e_3))t(e_3) + t(e_3) + h(e_3) \\
\text{By (6.5)} \quad & = (b_{\max} - \frac{b(e_1)t(e_1) + b(e_2)t(e_2)}{t(e_1) + t(e_2)})(t(e_1) + t(e_2)) \\
& \quad + t(e_1) + t(e_2) + h(e_1) + h(e_2) \\
& = b_{\max}(t(e_1) + t(e_2)) - b(e_1)t(e_1) - b(e_2)t(e_2) \\
& \quad + t(e_1) + t(e_2) + h(e_1) + h(e_2) \\
& = (b_{\max} - b(e_1))t(e_1) + t(e_1) + h(e_1) \\
& \quad + (b_{\max} - b(e_2))t(e_2) + t(e_2) + h(e_2) \\
& = f(b(e_1), t(e_1), h(e_1)) + f(b(e_2), t(e_2), h(e_2))
\end{aligned}$$

Which satisfies the homomorphism property.

Monotonicity: for simplicity, in the proofs that follow, the $h(e)$ terms have been neglected. The conclusions drawn are valid when the $h(e)$ are also included.
To satisfy equation 6.6, if $b(e_1)t(e_1) = b(e_2)t(e_2)$ and $t(e_1) = t(e_2)$ then clearly $b(e_1) = b(e_2)$. Hence, $f(b(e_1), t(e_1), h(e_1)) = f(b(e_2), t(e_2), h(e_2))$.

For equation 6.7, then the data transferred is equal but $t(e_1) < t(e_2)$. Given this, the aim is to prove that

$$\begin{aligned}
f(b(e_1), t(e_1), h(e_1)) & < f(b(e_2), t(e_2), h(e_2)) \\
\equiv (b_{\max} - b(e_1) + 1)t(e_1) & < (b_{\max} - b(e_2) + 1)t(e_2).
\end{aligned}$$

As $b(e_1)t(e_1) = b(e_2)t(e_2)$, the target of the proof can be changed to be

$$b_{\max}(t(e_1) - t(e_2)) + t(e_1) < t(e_2)$$

Given that $t(e_1) < t(e_2)$ it can be seen that this expression will hold.

For equation 6.8 the goal is to prove that

$$\begin{aligned}
(b_{\max} - b(e_1) + 1)t(e_1) & < (b_{\max} - b(e_2) + 1)t(e_2) \\
\equiv b_{\max}(t(e_1) - t(e_2)) - t(e_1) & < b(e_1)t(e_1) - b(e_2)t(e_2) + t(e_2)
\end{aligned}$$

Given that $t(e_1) \leq t(e_2)$, the left hand side will always be negative, whilst given that $b(e_1)t(e_1) > b(e_2)t(e_2)$ and $t(e_2) \geq 0$ the right hand side must be positive. Hence this equation holds.

Equation 6.9 is (unsurprisingly) a far more difficult condition to satisfy. The goal is to prove that

$$(b_{\max}+1)(t(e_1)-t(e_2))+b(e_2)t(e_2)-b(e_1)t(e_1)>0$$

under the specified conditions. If $t(e_1) > t(e_2)$ then we cannot prove whether this expression holds or not. If, for example, $t(e_1) \gg t(e_2)$, whilst $b(e_1)t(e_1) \simeq b(e_2)t(e_2)$, then this inequality would hold. If the inequality is expressed as $\frac{t(e_1)}{t(e_2)} > \frac{b_{\max}-b(e_2)+1}{b_{\max}-b(e_1)+1}$, it can be seen that as the traversal time of edge e_1 increases (relative to that of e_2), the inequality can be satisfied by higher and higher ratios of $b(e_1)$ to $b(e_2)$. Conversely, if the traversal time of e_1 is quite similar to that of e_2, but $b(e_1) \gg b(e_2)$, it transpires that $f(b(e_1), t(e_1), h(e_1)) > f(b(e_2), t(e_2), h(e_2))$, which is *not* as desired. Clearly, therefore, the proposed metric does not entirely satisfy the monotonicity condition, and hence will not provide routes that are globally optimal according to the conditions previously specified.

6.3.4 Near-Optimal Solutions

As seen in Section 6.2.2, no efficient method exists for optimally solving the multi-criteria routing problem. Instead, *efficient* (non-optimal) solutions are found. Many of the methods for efficient solution search are very resource intensive. Therefore, given that the family of metrics proposed in Section 6.3 satisfies the majority of the criteria required for it to be maximisable, and that a standard, efficient shortest path routing algorithm can be used, this has been chosen for routing over the multi-planar graph proposed in Chapter 5. Other multi-criteria routing strategies could equally well be used instead.

Provided that using the proposed weighting function produces results that are *better* than current approaches, it is believed that it is a useful formulation. This is a similar pragmatic approach as to that taken in the widely used Enhanced Interior Gateway Routing Protocol (EIGRP), which is known to have a non-monotonic metric [91], and hence suffers from a similar issue of not always giving the optimal route. This does prevent it from being considered as being very useful.

6.3.5 Comparison with Previous Approaches

In order to ascertain whether the scheme proposed in this dissertation is useful (despite its not entirely satisfying the monotonicity condition), it must be compared to what is currently available. The only existing (implemented) approach to ascertaining what networks should be used on a journey between two geographical points is to pick a single geographical route, (which is assumed to be the shortest distance route), and attempt to formulate a handover strategy over it. This approach is taken in Mobisteer [171], where the vehicle connects to whichever network was

historically recorded as strongest, regardless of the length of time that network was available for. Because of this, whilst it is a definite improvement over a purely reactive scheme, it will not compute the optimal sequence of handovers, even over the given geographical route. Moreover, it does not take into account the handover costs when choosing a network to connect to. In contrast, the novel approach given here of constructing a multi-planar graph and routing over it means that such factors can be taken into account *on multiple possible routes*. Whilst the choice of weighting function will mean that the route selected will not always be the optimal one, it is clear from the example given in Figure 5.1 and Table 5.1 that a near-optimal solution is highly advantageous compared to solely using the shortest path route.

A related (but unimplemented) solution given by Nickerson [173] involves setting the routing metric to be $f(e) = t(e) - \frac{b(e)}{b_{\max}} t(e)$. The aim is to transfer a *given quota* of data in the least possible time. In order to do this, the assumption is made that there is a hotspot with throughput $b_{\max}$ at the end of the journey, and the total journey time includes any time to be spent at the destination whilst any remaining data from the quota is transferred using the hotspot. However, this approach only allows a vehicle to make a stop at the end of its journey. If instead the hotspot were not at the destination but somewhere midway, it would be more time-effective to temporarily stop at the midway point. A concrete example is a journey between two rural locations that do not have connectivity, that traverses a city that has a large number of hotspots present. Here, it would be beneficial to park in the city for a short time, in preference to doing so at the destination. Whilst "transfer a given amount of data by arrival time" was one of the use cases in Section 5.1.2, the author, like Nickerson, has not currently found a solution that successfully achieves this goal with sufficient flexibility concerning the placement of high throughput networks. Finding an optimal solution to this class of problems is $\mathcal{NP}$-hard, as illustrated by Garey and Johnson [84], who proved that finding the *existence* of a path between two given nodes that had length and weight lower than two given values was $\mathcal{NP}$-complete.

Neither the method proposed here, nor Nickerson's method are able to express the possibility that a user might wish to travel along the same edge twice in order to achieve a better data transfer. The fundamental reason for this is that shortest-path algorithms deliberately avoid loops, as there is no way of specifying a limit to the algorithm as to how many times to go round such loops before exiting them. Similarly, it is not possible to easily express the possibility that it might be best for the user to make a short stop at a particular point along the route, if this will mean that the vehicle could remain in a high throughput coverage area for a short time in order to achieve the user's data transfer quota.

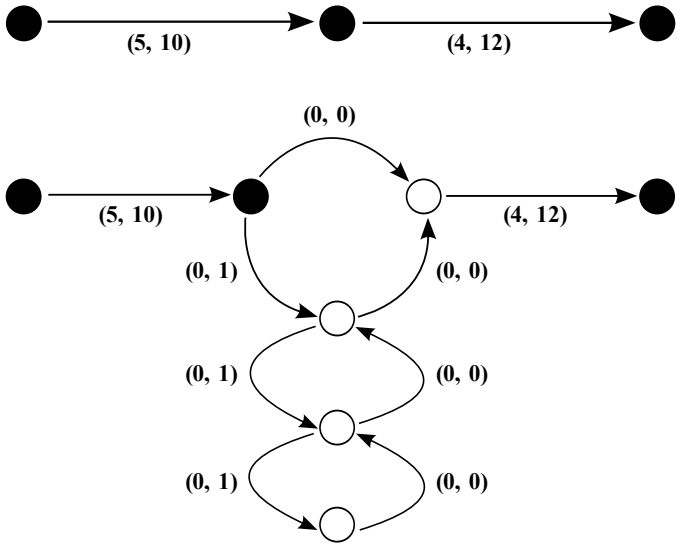


Figure 6.3: Partial loop unrolling, allowing up to a given maximum of time spent stopping at a particular point. The upper diagram shows the original graph, with edges annotated with tuples of edge length and traversal time. The lower diagram shows how a stop of up to 3 time units could be included in the route if loop unrolling (which creates the white nodes in the graph) is used.

One method for partially solving these problems is to use (partial) loop unrolling. To represent stops at a particular position of multiples of (say) one minute, firstly those areas where stops are both possible (e.g. car parks but not highways) and beneficial (in terms of data transfer) must be identified. For each of these locations, edges are created that have zero length, but a traversal time of some multiple of minutes (up to a given threshold), and consequently also have a particular amount of data transferable. Figure 6.3 illustrates this method. Theoretically the same technique could be used to represent multiple traversals of loops in the road network. However, this becomes more complex, as there are likely to be many exits of such loops (rather than only one entry and one exit, as with stops).

Therefore, given that no existing work uses a better method, the family of metrics proposed in Section 6.3 will be evaluated over the multi-planar graph. To ensure that all parameters used for the evaluation were realistic, the applicable throughputs and handover delays were carefully considered. The reasons for these choices are described in the next Section.

6.4 Throughputs & Handover Delays

In constructing multi-planar graphs two assumptions have been made that must be realistically quantified/explained:

- Approximate expected network throughput at a particular geographical location can be derived from sufficient RSS readings obtained at that point (Section 3.2).
- Disconnection times (or periods where throughput is impacted) incurred when handing over between different networks have well defined lengths.

These are carefully examined below.

6.4.1 Effects on Throughput

This dissertation has concentrated on utilising IEEE 802.11b/g and UMTS cellular networks. Both of these technologies can run at a range of bit rates that depend on the quality of the radio channel between the transmitter and receiver, as described in Section 2.3.3. For both of these technologies, it is important to note that the greater the number of users of the network, the lower the available network resources per user are. Also, with UMTS cellular networks, the coverage area of a cell decreases somewhat depending on the number and location of its users ("cell breathing"). For simplicity, in this work it is assumed that these effects are negligible. However, cell loads are likely to follow well known patterns (e.g. many users during the morning rush hour), and hence such information can be incorporated into coverage maps in order to provide a better predictor of achievable throughputs. Network traffic's packet size is also a factor that affects throughput; this detail is also neglected, but it should be noted that such performance effects have been well investigated [187]. Therefore, were the type of traffic to be sent known when routing was performed, the edge weights could easily be adjusted according to its payload size characteristic. Finally, it is assumed that the majority of traffic will be in the download (to the client) direction, and hence the throughputs and delays in the upload direction are not examined.

As an aside, the majority of wireless networks have an access network rôle, i.e. they feed data to and from a better provisioned core that is generally wired. Hence, whilst end-to-end protocol-level throughputs *are* affected by traffic travelling on the core network, the bottleneck will tend to be the bit rate achievable over the wireless link. Therefore, the assumption is made that the end-to-end throughput using a particular wireless network will depend overwhelmingly on the radio channel quality that the wireless network experiences.

6.4.2 RSS to Throughput Conversions

The data collected by the Sentient Van 3.1 includes RSS readings for a single provider's UMTS (HSDPA) network, and SNR readings for all 802.11b/g networks encountered. These can then be used to predict what throughputs will be achieved at any point. Brief experiments were carried out to ascertain the typical approximate TCP throughputs achievable on the live cellular network. These involved the downloading of a file from our laboratory web server over the public Internet using the Linux wget utility. The resulting conversions for UMTS are given in Table 6.1.

For converting from SNR to TCP throughput for 802.11g WiFi, a similar *small* set of experiments were performed. The only work relevant to this that could be found was by Na *et al.*, and this concerned only 802.11b [168]. The tests were carried out using a single AP on an otherwise unoccupied radio channel using the netperf[1] tool. In addition, the equipment was configured to assume a pure 802.11g (rather than a mixed 802.11b and 802.11g) environment. For mixed environments throughputs would be lower due to an RTS/CTS sequence needing to be carried out at 802.11b throughputs prior to each 802.11g data packet being sent. The resulting conversions are given in Table 6.2.

Two points are of note. Firstly, in these results, TCP throughputs are far lower than the link-layer throughputs that might be expected; for example, 802.11g has a theoretical maximum raw throughput of 54 Mbit/s, but here a maximum of only 20 Mbit/s is achieved. Such high link-layer throughputs are never seen at the TCP layer due to overheads such as framing at the layers below. Hence, these results should not be interpreted as especially low. Secondly, in a real deployment it is likely there would be some APs configured for mixed 802.11b and 802.11g environments, however, it is also true that in real deployments there will also be non-HSDPA cellular coverage areas, in which only sub-1 Mbit/s throughputs are possible. For simplicity, the highest throughput (but currently deployed) flavours of both WiFi and UMTS are used in the conversions given here. This has no impact on the evaluation of the *validity* of the proposed algorithms, and indeed for a real deployment it would be simple to add conversions for mixed environments to the implementation.

RSS (dBm)	Throughput (Mbit/s)
>-77	1.28
-86 .. -77	1.20
-100 .. -87	1.08
-111 .. -101	0.32
<-111	0

Table 6.1: Conversions used from UMTS RSS to TCP throughput.

[1] `http://www.netperf.org/`

SNR (dB)	Throughput (Mbit/s)
>30	20
20 .. 30	12
5 .. 20	8
<5	0

Table 6.2: Conversions used for 802.11g SNR to TCP throughput.

Fortuitously, these inexact conversions do not affect the validity of the results. This is because the aim of this work is to evaluate whether the proposed system to perform routing over a multi-planar graph (the proactive approach) results in better connectivity for a user than using a purely reactive approach. In the system's evaluation, both approaches make use of the real RSS values that were encountered in the real traces to calculate their respective mean throughputs and transfer sizes. Hence, the conversions are equally (un)biased for both approaches, and the evaluation is fair. The only caveat to this is that the magnitudes of the conversions must be correct: for example, were the HSDPA and 802.11g throughputs to be approximately equal, there would be little point in disconnecting from the cellular network and utilising WiFi hotspots. By obtaining approximate experimental results it has been ensured that the orders of magnitude are correct. Evidently, were this system to be deployed in a real-life scenario, further experiments to determine the exact conversions would need to be carried out.

6.4.3 Characterising Handover Delays

Previous work has analysed the delays that occur when a handover takes place. The delay can be split into the detection time (to receive the first signal from the network to be handed over to), client configuration time, registration (association) time, and the time for TCP (or other protocol) to adapt to the different throughput of a new technology (Section 2.6.2). For simplicity, it is assumed that the detection time is negligible, as in this scheme a handover is only begun when a reading from the target network is obtained. Client configuration time (of the order of one second) is also neglected. The remaining two stages of delay are the most significant: previous work on WiFi registration times [85, 72] and handover delays in general [234, 43] has characterised them for the handovers from GPRS to WiFi. Throughputs (of the source technology as compared to the target technology) are more similar for the UMTS HSDPA to WiFi case than the GPRS to WiFi case, but it is also of note that previous work used 802.11b for the WiFi tests, which has a lower throughput than 802.11g. Thus, the author has elected to decrease the adaptation time only marginally (rounding down to the nearest second) for the case of HSDPA to WiFi. It is also assumed that most traffic will be in the download direction, and hence the upload case is not examined, which would have slightly different adaptation times as cellular links (at present) are asymmetric in terms of

Source	Target	Delay Type	Length (s)
802.11g	HSDPA	Registration	3
		Adaptation	2
		Total Delay	5
HSDPA	802.11g	Registration	4
		Adaptation	0
		Total Delay	4
802.11g	802.11g	Registration	4
		Adaptation	0
		Total Delay	4
HSDPA	HSDPA	Registration	1
		Adaptation	0
		Total Delay	1

Table 6.3: Handover delay lengths.

throughput. Another assumption made for simplicity is that there is only a single operator's cellular network present, and hence the horizontal handover delay for HSDPA will only be incurred after a brief outage where the client loses signal, and hence the delay will be short. Therefore, the fact that for a horizontal handover to a *different* operator's network a longer registration delay would be caused is neglected. Table 6.3 gives the figures used for both types of handovers.

6.5 Evaluation

In order to evaluate the proposed proactive routing algorithm, it was necessary to compare it to the performance that could be obtained with a purely reactive approach, as evidently if there were to be no gain in using this more complex technique it would be a waste of resources. A non-trivial reactive algorithm was therefore created that would allow the advantages and deficiencies of the proposed proactive approach to be measured.

6.5.1 Reactive Algorithm

The reactive algorithm is run on traces of previously collected RSS data, (though there is nothing preventing its use in real-time). It is supplied with timestamped RSS data, one file for each network technology, each of which may then in turn include data for multiple administratively separate networks of the same technology (e.g. multiple WiFi ESSIDs). At each time step (set to one per second), the algorithm examines the readings up until that point. A decision is made concerning which network to utilise based on a small amount of historical data and the current

reading for each separate network. Concretely, a network is only considered viable to connect to if there have been at least k readings for it within a rolling window of length p seconds, where k is dependent on the expected time interval between readings, i, and is given by $k = p/i$. In this work, a rolling window of $p = 12$ seconds, an interval between WiFi readings of 2 seconds, and an interval of 6 seconds for HSDPA are used. This results in a requirement for 6 readings in every 12 seconds for a WiFi network to be considered viable, and 2 readings in 12 seconds for an HSDPA network to be considered viable.

In addition, the algorithm takes into account how the RSS for each network is changing, i.e. whether the trend is upward or downward. The handover decision algorithm will pick an upward trending network in preference to a downward one. However, if all networks that satisfy the k readings in the rolling window criterion have downward trends, the algorithm will pick the one whose RSS value implies the highest throughput. Trending is determined by examining the last eight readings for that network: each pair of readings has an upward or downward change. If the majority of the changes in the last eight readings are downwards, the trend is deemed to be down, otherwise it is up (i.e. if there is no majority an upward trend is assumed). All of these aspects are described in pseudocode in Algorithm 1.

By incorporating these two criteria into the reactive handover decision algorithm, connections to networks that are only briefly available are avoided, or to those that will quickly become unavailable (having a downward trend in RSS). However, this is of course traded-off against deciding to immediately connect to a potentially useful network as soon as it is detected. The values used for the parameters given above are, in the author's opinion, a realistic compromise.

Sometimes the reactive algorithm's choice will be in favour of a network that was viable in the recent past, but is temporarily unavailable (e.g. due to a building being in the way of the transmitter). If a reading for the network in question has not been obtained for 6 seconds it is assumed that it is not available. If a short time later it becomes available once more then a horizontal handover is assumed to have taken place, i.e. that there is a time cost incurred in re-registering with the network concerned. This is similar to the concept of using a zero-coverage plane in the multi-planar graph for a short period of time.

When the algorithm executes a handover, the relevant total handover time from Table 6.3 is treated as time spent disconnected. This is a conservative assumption, as during the adaptation time of a vertical handover throughput will be non-zero, but ensures that any TCP recovery time is accounted for.

Algorithm 1 CHOOSENETWORK(t) Returns the network to connect to, c

Require: t, the current time

$p = 12$ // Window size
$s = 1$ // Time step
$r = 8$ // Number of readings to trend over
N // Set of tuples of networks and when last seen
$R \leftarrow$ GETREADINGS$(t - s, t)$ // Get readings occurring between two times
for all $(n, s) \in R$ **do**
    UPDATE$(N, (n, s))$ // Update when each network was last seen
end for
$Q \leftarrow \emptyset$
for all $n \in N$ **do**
    $k_n \leftarrow$ GETKTHRESHOLD(n) // Get threshold for this network
    if NUMREADINGS$(n, t - p, t) > k_n$ **then**
        $Q = Q \cup \{n\}$
    end if
end for
$T = \emptyset$
for all $q \in Q$ **do**
    if TREND$(q, r) =$ up **then**
        $T = T \cup \{q\}$
    end if
end for
if $T = \emptyset$ **then**
    $c \leftarrow$ GETMAXTHROUGHPUT(Q) // Find the network of greatest throughput
else
    $c \leftarrow$ GETMAXTHROUGHPUT(T)
end if
return c

6.5.2 Comparison Methodology

To compare the two algorithms against each other, data from the Sentient Van (Section 3.1) was used, in conjunction with the extent generation algorithms given in Chapter 4. Map data from the Open Street Map project[2] enabled a coverage map of the local area to be generated and stored in a database. Further traces of RSS data were then collected using the Sentient Van and used as input to the two algorithms. The evaluation traces were *not* used as part of the input data to the coverage mapping algorithms. This approach meant that both real wireless network data *and* movement traces were used for evaluating the proposed algorithms.

6.5.2.1 Geographically Constraining the Multi-Planar Graph

For the proactive approach, instead of forming the complete multi-planar graph for the entire city, only that of the roads travelled on in the trace was constructed. This is termed *geographically constraining* the algorithm, as outlined in Section 5.1.2. To ascertain which roads these were, a map snapping technique similar to that presented in [121] was used. In this approach, two requirements must be satisfied. Firstly, the candidate road to which each point was to be snapped was restricted to only those that were reachable (in the graph) from the last point. Secondly, the distance between the current and last snapped points was required to be approximately equal to the distance between the corresponding raw GPS readings. The result was a robust algorithm that was successfully used to snap traces from the Sentient Van onto the Open Street Map data.

Having created the multi-planar graph using the roads obtained from map snapping, and the extents in the coverage map that corresponded to them, the routing algorithm could be run. The start node was selected to be the node nearest to the start of the trace and the destination the node nearest the trace's end point.

6.5.2.2 Updating Route Timings

By this stage we have a route that specifies the handover sequence that should be employed over the geographical path that was taken by the test vehicle (hereafter referred to as the *handover route*). However, this sequence assumes that the traversal times of the edges in the route are given by the speed limits on those edges. In practice, the vehicle is likely to have come to a stop at junctions or been slowed down in traffic. Therefore, it is necessary that the times of arrival at each node on the handover route be updated using the timestamped location information found in the GPS trace from the vehicle.

To perform this update, each junction along the handover route was examined, and its geographical location used to ascertain from the map-snapped GPS trace at

[2] `http://www.openstreetmap.org/`

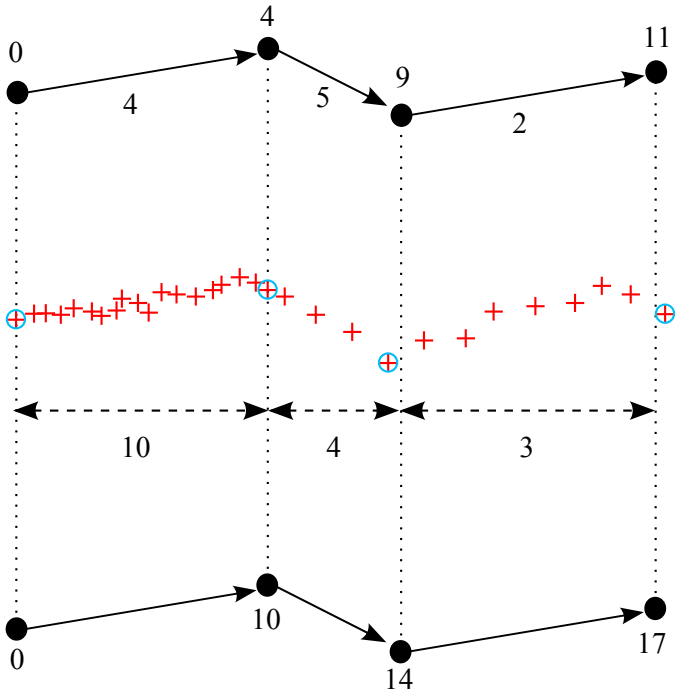


Figure 6.4: Edge traversal times on the proposed route (top) are updated according to the GPS trace (middle). The time taken to travel between each junction is found, and used to update the edge traversal times, giving an corrected route (bottom).

what time the junction was actually reached, as shown in Figure 6.4. This enabled the traversal times of each edge in the route to be updated. The end result was a handover route that incorporated the correct timings for each edge, thus enabling a realistic calculation of the network characteristics (such as total data transferred over the route) that the handover route predicted. One point to note is that *handover* edges did *not* have their traversal times updated. This is because such handovers take a fixed time, and are only initiated when the vehicle reaches a specified geographical location. If the vehicle travels slower than expected, its position will be different from that originally predicted by the handover route. This is actually advantageous: the vehicle will have travelled through *less* of the extent that it was handing off into, and hence will derive even *more* benefit from it than was expected.

6.5.3 The Need for Accurate Speed Data

One situation in which the route produced by the proactive approach may be non-optimal is when the actual time spent on a particular road is very much *greater* than its traversal time as recorded in the database (by default the road's length

divided by its posted speed limit). Such a situation might arise in heavy traffic that is moving slowly, or at traffic lights, which at some times may involve waits of up to a minute, and at others hardly any wait at all. In these cases, the proactive scheme might choose to ignore a high throughput network, as it assumes that such a network will only be briefly available. The reactive approach, meanwhile, chooses to utilise that network because it is available for a significant length of time. To mitigate the effects of congestion, road traversal times as stored in the database should be calculated from databases of average road speeds. Such information is already in use today in high-end satellite navigation units which update routes to allow for traffic conditions. The states of traffic lights along the route could also be incorporated to further improve accuracy. Additionally, a proactive algorithm could fall back to a reactive one when it detected that speeds were slower than predicted, or when no coverage was predicted to be available.

6.5.4 Retrieving Relevant RSS Values

The updated handover route must then be evaluated by using the timings to pick out the relevant RSS readings from the log of the trace taken by the vehicle. The algorithm goes through the handover route, calculating the arrival and departure timestamps for each edge, and retrieving the RSS readings that are between those times and are for the wireless network that is specified by that edge. These readings correspond to what a vehicle making use of the proactive technique would experience, and hence it is possible to compare the network characteristics that would be achieved using this method, as compared to the reactive method proposed above.

One caveat to this approach is that to make the comparison fair the reactive approach must be restricted to only using networks for which extents are present for in the coverage map. This means that wireless networks (particularly WiFi) that have only been observed on perhaps one or two previous traces are excluded from consideration by both algorithms. In practice, were a proactive scheme to be deployed, it would incorporate a reactive component, which would be used for portions of the route where the coverage map was not aware of any wireless networks. Such a reactive component could also be engaged when the speed of the vehicle is very much lower than expected (see above).

The route suggested by the proactive algorithm will sometimes include handovers to networks that are not encountered in the trace, e.g. an access point was available for less time than expected. A handover was only counted as having taken place when a reading for the network that the handover was to was obtained. If no readings for the target network were obtained, it was assumed that the client was disconnected from the time at which it was to have carried out the handover. This approach is similar to the reactive algorithm's, and hence ensured a fair comparison between the two. Hence, in the statistics concerning handovers the counts are for handovers that would have actually taken place, rather than all of the ones that were *suggested* in the proactive algorithm's route.

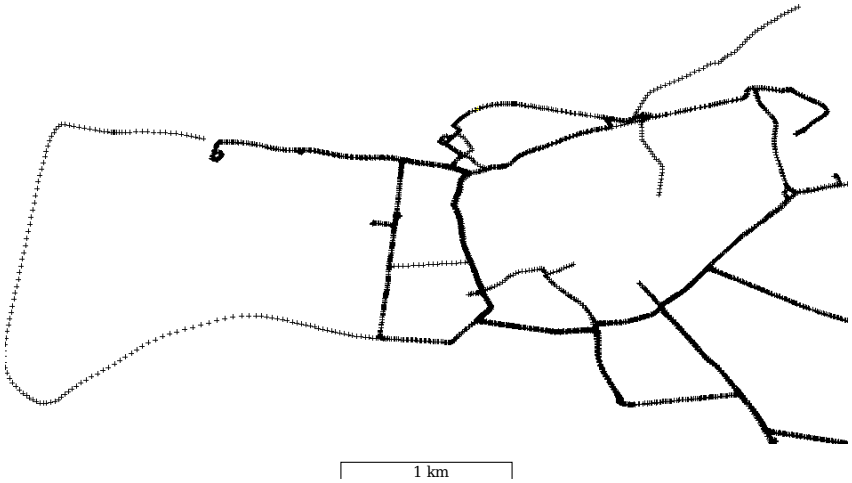


Figure 6.5: GPS data from all the traces used for evaluation, superposed on top of one another.

6.5.5 Routing Metrics

The steps described in Section 6.5.2 were performed for 25 traces, using six different weighting functions for each. In the equations that follow, b signifies the mean throughput over an edge (set to 10^{-11} if the throughput was zero), $b_{\max}$ the maximum possible throughput achievable with any technology on any edge (which was set to 25 Mbit/s, to bound the maximum 802.11g throughput that had been previously observed), t the time taken to traverse the edge, s the physical length of an edge, and h is either 1 or 0 depending on whether the edge is a handover one or not:

- **A:** Inverse throughput, $w = 1/b$
- **B:** Inverse transfer size, $w = 1/bt$
- **C:** Weighted combination of inverse transfer size and edge length, $w = \frac{10^{-1}}{bt} + s$
- **D:** Traversal time weighted by throughput difference, $w = (b_{\max} - b + 1)t$
- **E:** Weighted combination of difference in throughput, time, $w = ((b_{\max} - b)^3 + 1)t$
- **F:** Weighted combination of difference in throughput, time, and handover penalty, $w = ((b_{\max} - b)^3 + 1)t + 10^3 h$

The weightings used above, such as 10^{-1}, were set to ensure that every term in each equation has a similar order of magnitude size. The value for the handover

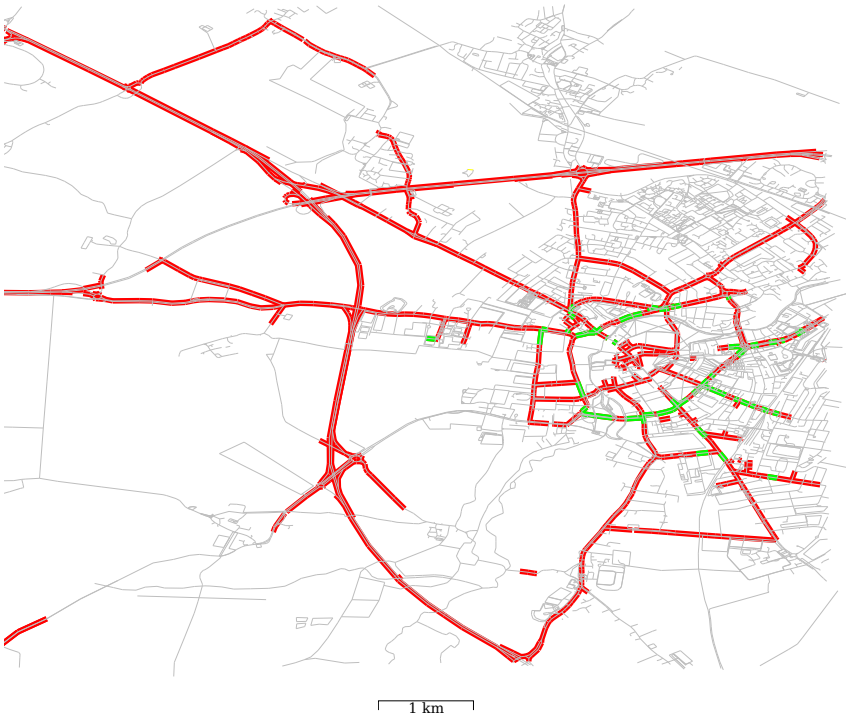


Figure 6.6: Roads included in the coverage map generated by Nearest Neighbour interpolation. Red indicates UMTS coverage, green indicates WiFi coverage. [OSM].

penalty was set to 10^3, so that it was of similar magnitude to the data transfer term. This value was found to reduce the number of handovers considerably, whilst still allowing the more worthwhile ones to occur.

The first weighting function illustrates how a very naïve approach might select networks to handover to, if provided with a particular set of roads that are to be travelled on. In this scheme the algorithm greedily selects all of the highest throughput edges, regardless of their length (and hence the quantity of data that could be transferred). This metric is not suited to the more general case where the algorithm is requested to provide the optimal route between two specified points. This is easily illustrated by considering a short journey between two points in a rural area. If routing by distance or time we are likely to take the most direct route, through an area of (probably) poor network coverage. If routing by inverse throughput (metric A), we might be recommended a route that traversed every street of every city in the country, in order to connect to as many WiFi hotspots as possible. Evidently, therefore, for generalised routing the metric must take into account one or both of edge traversal time or edge length.

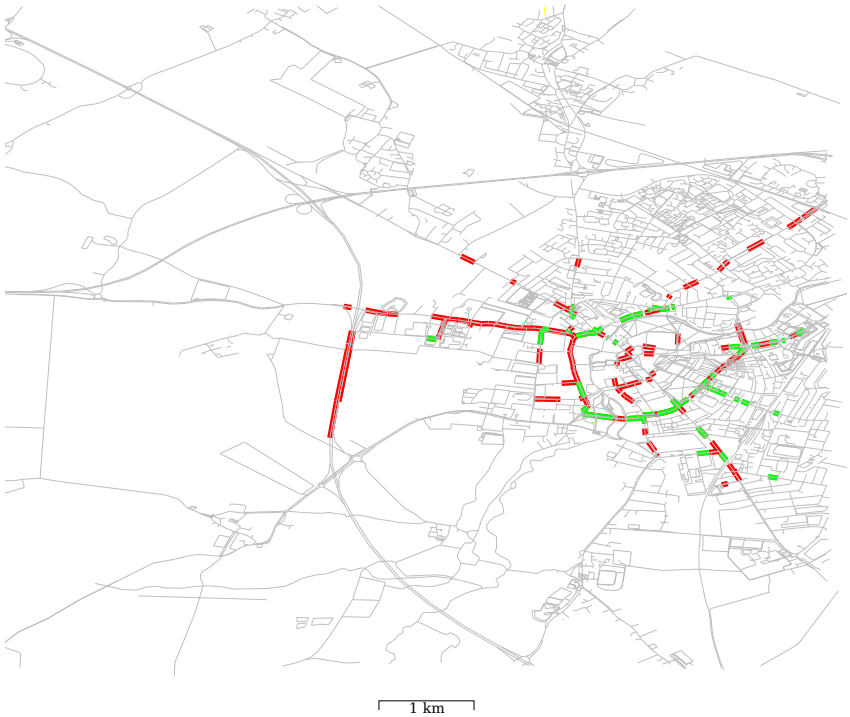


Figure 6.7: Roads included in the coverage map generated by density-dependent dominant point smoothing (requires a minimum of 50 points). Red indicates UMTS coverage, green indicates WiFi coverage. [OSM].

The second metric, B, weights an edge according to the amount of data that could be transferred as it is traversed. This approach is of particular interest as it penalises edges with no wireless coverage (their throughput is set to a value of 10^{-11}). The amount of data transferred does mean that edges are chosen on the basis of a more useful quantity than solely throughput. The latter might result in a very short edge of high throughput being selected, rather than a single edge of slightly lower throughput but of substantial traversal time, and hence data transfer.

Metric C combines the inverse of the amount of data transferable over an edge with the edge length. This then means that whilst the metric decreases as more data is transferred, it increases with distance. Thus it does achieve the goal of trading-off these two quantities, though it does not satisfy the homomorphism property given in Section 6.1.

Metric D utilises the formulation of the weighting function proposed at the end of Section 6.1. This allows data transferred and edge traversal time to be offset against each other. Here, the simplest form where $n = 1$ is used, in order to compare the performance where the metric satisfies the homomorphism property to the performance of metrics where $n > 1$ (and which therefore do not satisfy this property).

Metric E increases the value of n to be 3, placing more emphasis on the amount of data transferred as compared to that which would have been done so had a throughput of $b_{\max}$ been present. Results are not given for $n = 2$, as the results obtained in Section 6.7 showed this (and $n = 4$) performed less well than $n = 3$.

Metric F augments this approach by introducing a penalty against carrying out handovers. This allows the metric chosen to be application-mix dependent. If there is a real-time media application in use, the user is likely to prefer fewer handovers in order to have fewer interruptions to the stream. Handover delays are traded-off with the greater data transfers that might be possible on other networks, which would be useful for disconnection-tolerant applications.

6.5.6 Results

The 25 traces used for the evaluation are shown in Figure 6.5. In total, these traces contained over 5,000 GPS locations, and were driven over the course of July of 2007. Some roads were traversed more than once, whilst others are included in only one trace. The driving was carried out by members of our laboratory for their normal tasks, i.e. *not* solely for the purposes of this experiment. The driving behaviour is therefore not biased for or against this evaluation. Note that we do not have sufficient data recorded for *all* the roads depicted in Figure 6.5, and hence there are many that are not present in our coverage map, as shown in Figure 6.7. For evaluating both the reactive and proactive algorithms, the input traces were purged of all readings for networks that were on roads which were not included in the coverage map. This resulted in a high percentage of time spent disconnected in each trace.

The performance of the six proactive metrics and the reactive algorithm was evaluated by comparing the mean throughputs achieved over the traces, the percentage of the total trace time that the vehicle spent without a connection to any network, and the number of handovers performed throughout the trace. The results obtained for each metric are shown for each of the 25 input traces in Figures 6.8, 6.10 and 6.12. For space saving reasons the quantity $b_{\max} - b$ has been abbreviated to B, and the weighting of 10^{-3} on h in metric F has been omitted in the graph legends. In addition, a quantitative measure of how successful each proactive algorithm was as compared to the reactive algorithm is provided in Table 6.4. This gives the percentage of the 25 input traces where each proactive algorithm performed better than the

Metric	h	$\max(\bar{b})$	$d\%$
A	28.0	20.0	40.0
B	20.0	24.0	60.0
C	44.0	20.0	12.0
D	0.0	76.0	12.0
E	0.0	76.0	12.0
F	16.0	72.0	20.0

Table 6.4: Percentage of traces achieving better results of each type (column) than the reactive algorithm, by metric. h is (fewer) handovers, $\max(\bar{b})$ is (higher) mean throughput, $d\%$ is (lower) percentage of total trace time spent disconnected.

reactive one. Finally, in order to assess by what factor the proactive metrics improve performance, relative to the reactive algorithm, Figures 6.9, 6.11, and 6.13 are given. For each metric, each trace is represented by one point, which plots the difference between the proactive and reactive algorithms as a fraction of the reactive algorithm's result. For example, for the $1/b$ metric, the number of *extra* handovers carried out compared to the reactive algorithm, divided by the number carried out by the reactive algorithm, gives the improvement factor. The bars depict the mean improvement, with 95% confidence intervals shown. It is important to note that for the handover and disconnection time plots, a *negative* improvement factor indicates fewer handovers, or a lower disconnection time, for the proactive metric as compared to the reactive algorithm.

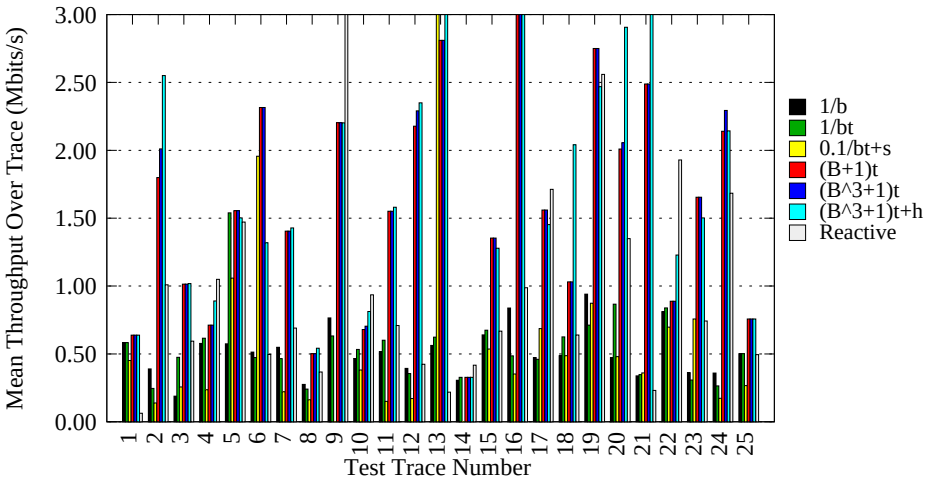


Figure 6.8: Mean throughputs achieved using each proactive metric, and the reactive scheme. Y-Scale truncated for clarity.

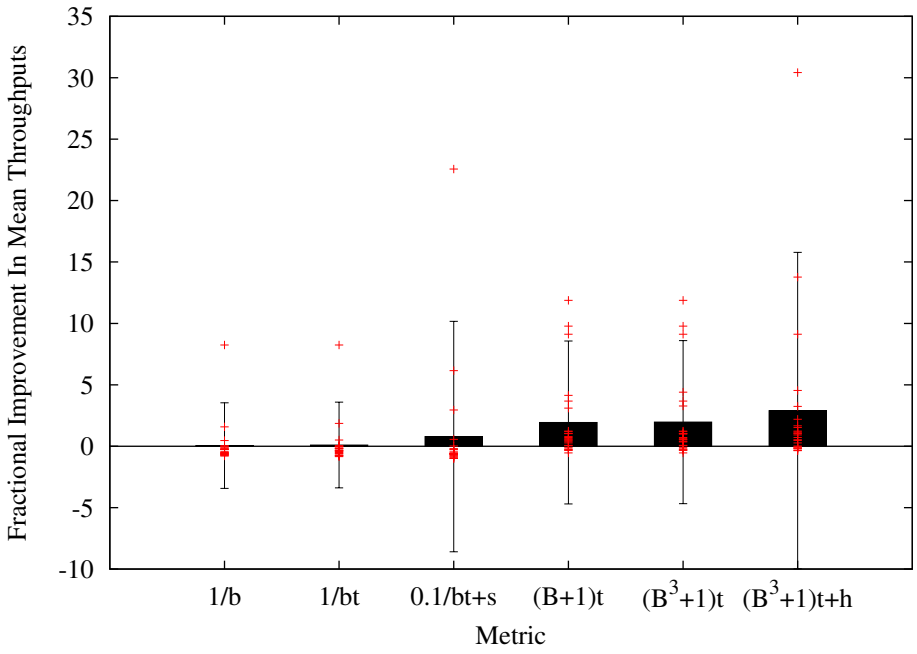


Figure 6.9: Fractional improvements in mean throughputs achieved using each proactive metric relative to the reactive scheme. Bars depict mean improvement, with 95% confidence intervals, points are individual trace improvements.

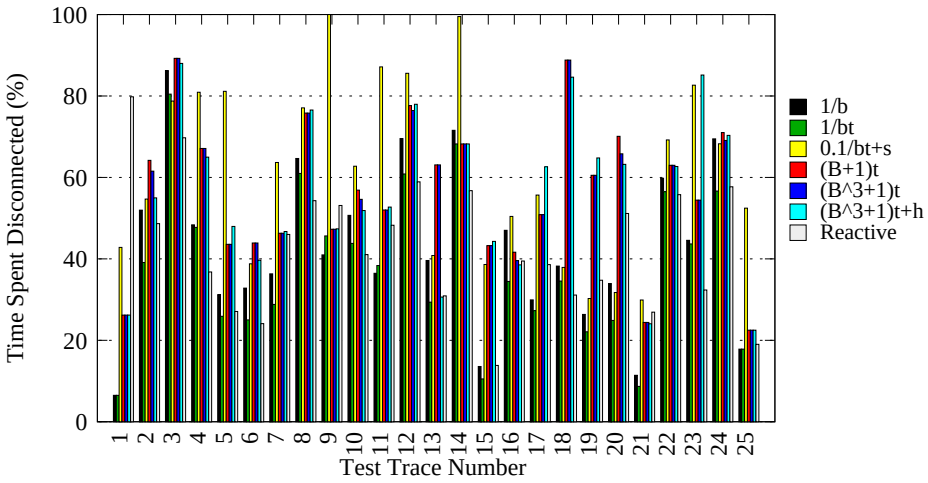


Figure 6.10: Percentages of time spent disconnected using each proactive metric, and the reactive scheme.

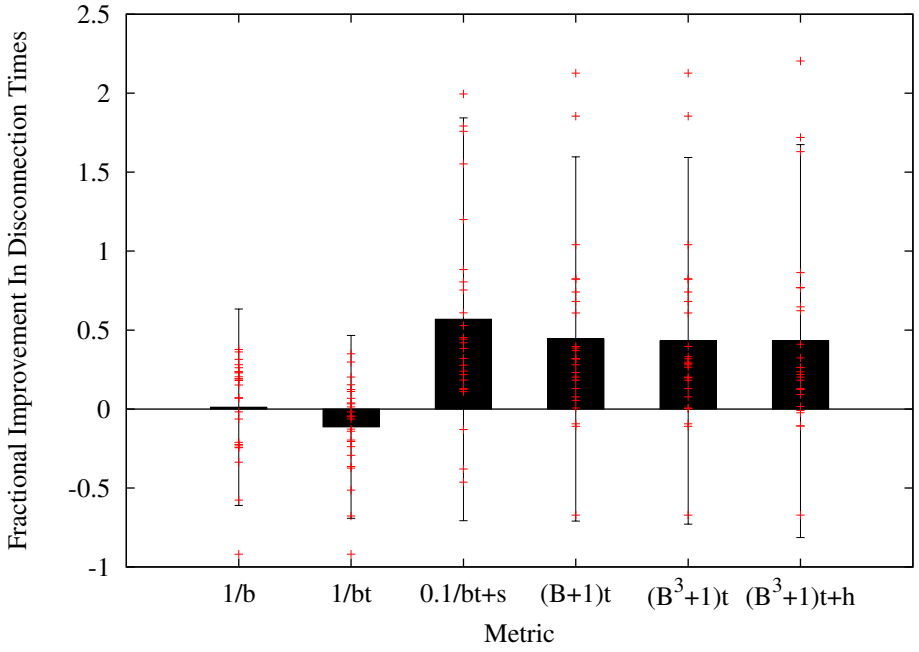


Figure 6.11: Fractional improvements in percentage of time spent disconnected achieved using each proactive metric relative to the reactive scheme. Bars depict mean improvement, with 95% confidence intervals, points are individual trace improvements.

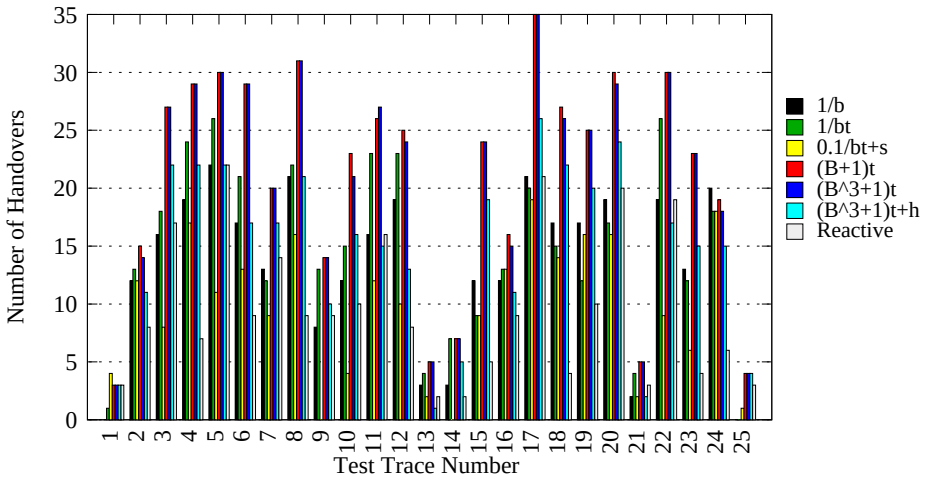


Figure 6.12: Number of handovers experienced (including to and from zero-coverage planes) using each proactive metric, and the reactive scheme.

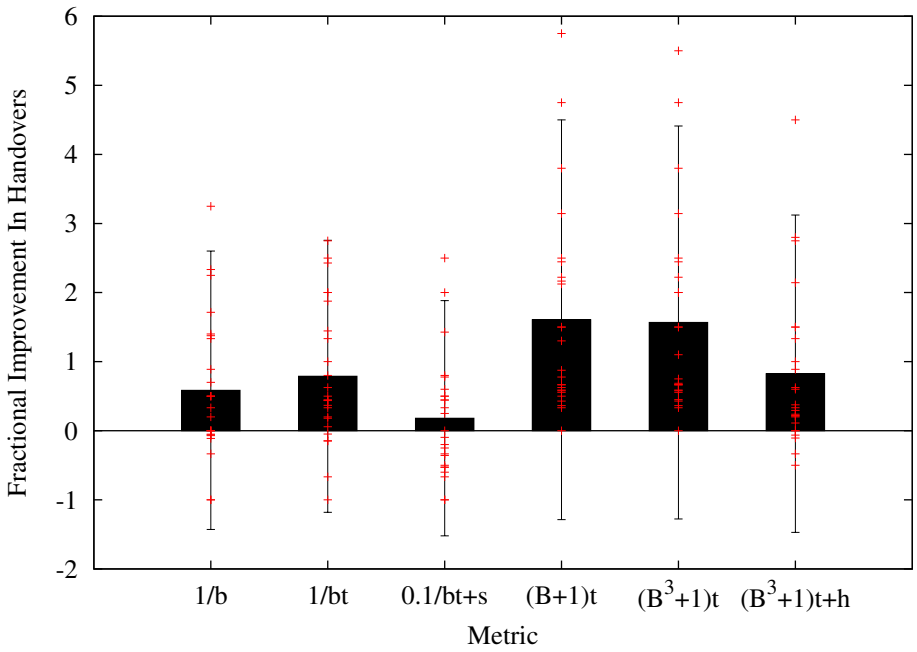


Figure 6.13: Fractional improvements in numbers of handovers achieved using each proactive metric relative to the reactive scheme. Bars depict mean improvement, with 95% confidence intervals, points are individual trace improvements.

6.6 Discussion

6.6.1 Mean Throughput

Mean throughput for each trace was calculated by assuming that each RSS reading from the trace would apply for a minimum of one second, and calculating the data transfer achieved if that RSS value was converted into a TCP throughput[3] (as given in Tables 6.1 and 6.2). These transfers were summed over the entirety of each trace, and divided by the trace's total time length. Hence, the mean throughput includes those times when the client would have had no connectivity, i.e. was disconnected.

Metric A, inverse throughput ($1/b$), was somewhat erratic in its performance in terms of the mean throughput. This is not a particular surprise since it neither conforms to the homomorphism property, nor includes any notion of edge traversal time, and hence will not take into account *how long* a high throughput edge is actually available for. Compared to the reactive algorithm it achieved a higher mean throughput in only 20% of the test cases: a poor performance.

Metric B, inverse data transfer size ($1/bt$), was even more erratic, though it did perform better than the reactive metric in 24% of the test cases. In test cases 1, 5, and 22 its performance was among the best exhibited, but in the majority it performed very poorly. Hence, it is clearly not a reliable metric from the point of view of achieving high mean throughputs. This is not a surprise given that the metric does not conform to the homomorphism property, and hence the weightings of two routes that do in fact have equal mean throughputs and times will not necessarily be equal, thus resulting in this seemingly random behaviour.

Metric C, inverse data transfer combined with distance ($\frac{10^{-1}}{bt} + s$), performed equal worst (with metric A) compared to the reactive algorithm (20%), and was worst of all the metrics for mean throughput in 17 out of the 25 test cases. For similar reasons to metric B, this variable performance is not surprising, but it is interesting that its mean throughputs are so different from metric B's, solely due to the inclusion of the distance term. Metric C performs best in terms of fewest handovers, but as a result does not connect to some of the more useful networks, thus causing long periods of disconnection compared to the other metrics.

Metrics D, E, and F, variations on the formulation that was suggested for the weighting function in Section 6.1, all performed similarly well. Over 75% of traces showed higher mean throughputs compared to the reactive algorithm for D and E, and 72% for F. Crucially, D, E and F performed consistently well, i.e. they did not exhibit the erratic behaviour seen for metrics A and B, an important factor when considering whether there would be any worth in deploying such a proactive

[3] For simplicity, the fact that TCP throughputs begin at reasonably modest levels and ramp up as acknowledgements are received has been elided. This is in part taken into account by the allowance for TCP adaptation time that is included for certain types of inter-network handover.

scheme. Figure 6.9 emphasises these conclusions, with metrics D, E, and F having high improvements in throughput compared to the reactive algorithm.

In four traces (9, 10, 17, and 22), the reactive algorithm performed better than any of the proactive ones. The first part of trace 9 concerns an approach to a busy junction, with the vehicle travelling much slower than the expected speed for that road. When stopping at traffic lights, it was in the vicinity of one particular AP for much longer than it would have been, had it been travelling at the road's speed limit. The reactive algorithm was therefore able to make good use of this AP. In contrast, the proactive algorithms did not elect to connect to the AP, as they determined that at normal vehicular speed the time spent in its coverage would have not been long enough to justify the handover costs to/from it. This illustrates the need for a more accurate database of prevailing speeds, as well as the graph's edges taking into account the delays incurred when a vehicle traverses a major junction. Such augmentations are already present in commercial navigation systems, and hence could (were we to have sufficient junction information) also be applied to the system proposed here.

On close examination, the reactive algorithm performed well in trace 17 because an access point that is present in the coverage map having the ESSID "x7temp" was not seen on the drive for this trace. It is surmised that this AP had been removed in the time between the coverage map being constructed and the test trace being driven. This illustrates the need to maintain coverage maps up-to-date, though given this is the only AP that significantly impacted any of the traces, it does give an indication that the churn in deployed APs is relatively low. Many APs may be, however, *added* over time.

Traces 10 and 22 were interesting in that the reactive algorithm chose a different AP to the proactive ones, and achieved a higher mean throughput because of it. This identifies a deficiency of conservative approach to coverage mapping: occasionally the coverage of an AP may be extend further than normal. This might take place due to low traffic volumes (and hence fewer vehicles blocking the transmission path), or a door/window being open in the building the AP is located in. If this happens regularly, the coverage map will reflect it, but if it does not, the map should be conservative in its estimates. This means that on the rare occasion when coverage is better than expected, a proactive algorithm will disconnect from the network before it would be best to do so. Whilst traces 10 and 22 illustrate this, they are the only cases in the 25 where such an effect is evident. Moreover, the difference between proactive and reactive mean throughputs for trace 10 is minimal. Hence, the problem is of negligible importance.

6.6.2 Time Disconnected

All results shown in Figure 6.10 exhibited high percentages of time spent disconnected. The primary reason for this is that a relatively large number of the roads travelled on in the traces were not present in the coverage map (as shown in Figure 6.7), and hence the time spent on these roads is counted as disconnected. In reality there is likely to be more than sufficient coverage of multiple types on these roads, and the problem is simply that more RSS readings are required to generate a more comprehensive coverage map. Whilst this is not ideal, the results do still allow a comparison of the different algorithms and metrics in terms of how much longer they spent disconnected compared to each other.

Interestingly, the reactive algorithm achieved low disconnection times on many of the traces, and Table 6.4 and Figure 6.11 show how in the majority of the traces the proactive algorithms were not better in this respect. This can be explained by noting that the figures in the handover column of Table 6.4 are also low, indicating that the number of handovers carried out by the proactive metrics was generally higher than the reactive algorithm. This caused more time spent disconnected (in part due to transferring to zero-coverage edges in preparation for handover), but allowed connections to be made to networks that provided far better throughputs and hence quantities of data transfer. As more networks are included in the coverage map, the reactive algorithm's performance as regards time spent disconnected is likely to remain relatively constant, as each network added would only be available for a short length of road, and hence the reactive algorithm would be unlikely to connect to it for long before needing to switch to another. In contrast, the proactive metrics would take advantage of this greater number of networks, carrying out a greater number of handovers, but reaping the benefits of greater time spent connected to high throughput networks.

It is worth noting that metric B ($1/bt$) has many traces for which it achieved a low percentage disconnection time (in 60% of the traces it performed better than the reactive algorithm). This is due to it's preference for edges with higher quantities of data transferred, rather than solely higher throughput (metric A). Metric A takes no account of the *length* of the edges, and hence might select a high throughput edge that is short and after which another handover is necessary. Metric B, whilst carrying out a greater number of handovers, connects to those networks which will provide the best connectivity for longest.

Metrics C, D, and E had poor disconnection times compared to the reactive algorithm, being better in only 12% of the test cases. D and E fared better than C in 16 of the test cases, but never outperformed metrics A and B.

In contrast, metric F had a slightly better performance, with 20% of the traces being less disconnected than for the reactive algorithm, and a lower mean increase in disconnection times (compared to the reactive algorithm) than C, D, or E (see Figure 6.11). This is due to its having carried out fewer handovers than D and E,

but caused it to have a slightly lower number of cases where its mean throughput was better than the reactive algorithm.

6.6.3 Handovers

Metrics A and B had similar performances in terms of number of handovers carried out. In 16 out of the 25 test traces, A had fewer handovers than algorithm B. Meanwhile, in 28% and 20% of cases respectively they outperformed the reactive algorithm. Their low number of handovers compared to the other proactive metrics was due to their preference for avoiding zero throughput edges if at all possible. Hence, because handover edges had very high costs, they were chosen as rarely as possible in the routes produced by A and B. The result was few handovers but low throughputs, as discussed in Section 6.6.1.

Similarly, metric C had a very low number of handovers, as shown by a small bar in Figure 6.13, performing best compared to the reactive algorithm. This is also due to its having used inverse throughput as a term in its metric. As previously discussed, this resulted in a low mean throughput.

Metrics D and E had very large numbers of handovers, as they sought to increase mean throughput by means of handing over to useful networks as frequently as possible. In contrast, metric F, which included a handover penalty, performed better, with its number of handovers being comparable to those of metrics A and B in 17 of the 25 test cases. Interestingly, this achievement did not impact F's value for mean throughput significantly, suggesting that in some cases the penalty incurred by carrying out a handover was not recompensed by short-term gains in throughput. This is further borne out by some of the mean throughput results in Figure 6.8 (traces 2, 18, 20 and 21) that show F had higher mean throughputs than metrics D and E, and yet carried out fewer handovers. Such behaviour is due to metrics D and E having elected to handover to networks that were predicted to be available for only a very short time, and then in the real trace no or few readings were in fact obtained for those networks. Metric F, due to its inclusion of a handover penalty, did not choose these fleetingly available networks, as the handover penalty exceeded the small extra data transfer benefit of using such networks. Thus, it appears that a carefully set handover penalty benefits the user not only in reducing the number of handovers carried out, but prevents handovers to networks that are available for such short periods that there is a real possibility of not being able to utilise. Further investigation is needed to ascertain what the optimal handover penalty is.

6.6.4 Overall

Overall, metrics D, E and F proved to be consistently better than the other metrics, and (crucially) the reactive algorithm as regards mean throughput. This achieves their primary goal of improving the amount of data transferred. In addition, these metrics also performed well (though were not the best) when considering the percentage of time spent disconnected. As regards number of handovers, metric F was comparable to metrics A and B, whilst not substantially impacting throughput. This suggests that the usage of a handover penalty term works well.

It can therefore be concluded that the proposed formulation of $f(b(e), t(e), h(e)) = (b_{\max} - b(e))^n t(e) + t(e) + h(e)$, with the $h(e)$ term being weighted appropriately, provides a performance that is consistently better in terms of mean throughput than a reactive algorithm, whilst achieving a number of handovers that is lower than other proactive metrics, and is in many cases comparable to the reactive algorithm's number. This demonstrates that, with an appropriate metric, a proactive handover algorithm provides significant benefits, with gains in mean throughput of over 1 Mbit/s in many cases. Although the proactive algorithm (with any metric) spent a greater percentage of time disconnected than the reactive algorithm did, this disparity would not be present if the coverage maps used were to include extents for more of the roads driven on in the test traces.

6.7 Unconstrained Routing

Having evaluated the six metrics in the constrained trajectory case, (i.e. where a geographical route is pre-defined) the problem of unconstrained routing is now considered. Here, the multi-planar graph for the city as a whole is generated, and the routing algorithm is free to choose whichever roads it deems best, having a notion of both time and network performance.

A set of 20 test cases were generated by requesting a random selection of pairs of road junctions from the database, and using each of these to provide the start and end points for a test case. A total of 10 different metrics (including the six outlined in Section 6.5) were tested over each test case, allowing each metric to (potentially) select a different geographical route. Dijkstra's algorithm was also used to find the shortest (geographical) length path between each pair of points. Each of the 10 metrics was then re-run with their route constrained to this shortest path.

6.7.1 Target Throughput Metric

A metric that was not presented in Section 6.5 was one that attempted to find a route that satisfied a given minimum throughput. In this Section, a metric similar to metric D in Section 6.5 is evaluated, but that instead of using the maximum possible throughput $b_{\max}$ uses a target throughput b_{target} (and caps the minimum value returned by $b_{\text{target}} - b$ to be zero). This allows an application to specify what the minimum throughput it requires is, and the algorithm will attempt to find a route such that this target throughput is always achieved. Clearly it may not be possible to achieve this target on some portions of the route. In this case the algorithm will choose the edges with the least traversal time and distance. In the experiments the target throughput was set to be 1.2 Mbit/s. This ensured that it was at least plausibly achievable, given that HSDPA throughputs are regularly of this order of magnitude.

6.7.2 Choice of Coverage Mapping Algorithm

In order to compare the efficacy of each metric in calculating routes that were most beneficial, a large number of roads in the city were required to be included in the coverage map. This then made multiple different routes feasible (in terms of network coverage) between most pairs of points. The algorithm used for generating the coverage map used in Section 6.5 was density-dependent dominant point detection, which requires a minimum of 50 RSS input points on a road before it will give a meaningful result. In contrast, the Nearest Neighbour Interpolation algorithm for generating coverage maps has no such minimum bound, but produces less accurate results, as outlined in Section 4.8. The coverage maps given by Nearest Neighbour interpolation and density-dependent dominant point detection are shown in Figures 6.6 and 6.7 respectively, showing how the former covers more roads. Hence, in order to provide a meaningful evaluation of unconstrained routing, the map generated by Nearest Neighbour interpolation was used for the experiments described in this Section.

It is important to note that in order to evaluate the accuracy of proactive metrics based on coverage maps compared to a reactive approach (as described in Section 6.5), it was necessary to use the density-dependent dominant point detection algorithm, as in Section 4.8 it was concluded this was one of the two best of the algorithms tested.[4] In this Section, the goal was to evaluate how well unconstrained routing performed using the multi-planar graph approach, as compared to taking the shortest geographical route and running a proactive handover algorithm on the edges of that route alone. Therefore, for this part of the evaluation it is irrelevant

[4]Previously, it was concluded that the best algorithm was Savitzky-Golay smoothing followed by dominant point detection. This algorithm was not used in this Chapter because the number of roads it covers is too few for a meaningful evaluation to be performed. This is because it requires a minimum of 101 RSS readings per road in order to generate that road's extents.

how accurate the coverage map is; only that it is an approximate representation of real-life, and that it covers a large number of roads. Hence, the use of the coverage map generated by the Nearest Neighbour Interpolation algorithm does not affect the conclusions drawn concerning the performance of unconstrained routing.

6.7.3 Methodology

The results for these test cases were generated by assuming that a vehicle would travel along the roads in the database at the posted speed limits. The sequence of networks that the algorithms recommended, along with the time a vehicle would spend connected to each, was then used to calculate the data that could theoretically be transferred over the recommended route. The mean throughput over the route, the number of handovers that would take place along it, the total time taken to traverse the route, and the percentage of the total traversal time that would be spent disconnected were also evaluated. Table 6.5 shows the number of test cases where each metric performed best, out of a maximum total of 20. A row is given for each metric when used for unconstrained routing, and another for each metric when constrained to the shortest geographical path between the start and end point of that test case.

6.7.4 Results

Table 6.5 enables several broad conclusions to be drawn. Firstly, in terms of mean throughput, the greatest values are achieved by proactive metrics using unconstrained routing (i.e. the $\max(b)$ column has high figures in its top half as compared to its bottom half, as does the data transfer column $\max(bt)$). Secondly, in terms of which metrics had the minimum number of handovers ($\min(h)$), unconstrained and geographically constrained routing performed (approximately) equally well (with the exception of the $1/bt + s$ and $1/bt + t + h$ metrics). Thirdly, although in some cases unconstrained routing generated routes that were longer than the shortest path's traversal time, many are within 10% of this (see the $\min(t) \pm 10\%$ column).

More specific conclusions can be arrived at concerning certain metrics. In particular, the $1/bt+s$ and $1/bt+t+h$ metrics (metric C in Section 6.5), continued to show the least number of handovers, and hence performs badly in terms of disconnection times. See Section 6.6.2 for an explanation of this.

Also of note is that the metric that specified a target throughput of b_{target} had mixed performance in terms of how well that target was achieved. The $b >= b_{\text{target}} \pm 10\%$ column of Table 6.5 shows that most of the metrics did not achieve the target throughput for any test cases. The metric specifically intended for this purpose, $(b_{\text{target}} - b + 1)t$, achieved the target mean throughput of 1.2 Mbit/s (or above) in only 3 out of the 20 evaluation traces. The question therefore arises as to

why this success rate is so low. The most likely cause is that there was not a viable route between the start and end points of the remaining test cases where the route had high enough throughputs to satisfy the target. This result would improve, in part, if more RSS data were collected that could be incorporated into the coverage map. Also, using a value of $n > 1$ with this metric might also produce better results.

It is important to note that the throughput value used to decide whether the route was above or below the target b_{target} was the mean throughput over the entire trace. Therefore, even if the mean throughput exceeded the target, this does not give an indication of how much of the route was below target (and was compensated for by other parts of the route being above target). Improvements to this metric are left as further work.

In terms of which other metrics achieved the target a substantial number of times, it is interesting that when run on the shortest path routes many metrics improved (i.e. achieved within 10% of the target throughput in more of the test cases). This is most likely to be because the shortest path routes would be the those along arterial routes that are normally taken when the Sentient Van is used. Hence, such shortest paths would have the most raw RSS data collected on them, and therefore be most likely to have continuous coverage in the coverage map. This is *not* to say that such a route would have the best (e.g.) mean throughput, which is why other routes are chosen in the unconstrained cases. However, it *does* mean that metrics that are heavily biased against zero coverage (i.e. $1/b$ and $1/bt$ and any metrics having terms of this form) will choose routes with continuous coverage, and hence be more likely to achieve the target. When run unconstrained, other metrics will select routes to attempt to maximise data transferred.

The most important conclusion that can be drawn from these results is that the proposed metric of $(b_{\max} - b)^n t + t$ performed well, with a value of $n = 3$ appearing to be marginally better than $n = 1$ for data transfer being within 10% of the maximum achieved, and $n = 1$ performing well otherwise. As previously noted, the value of n used will depend on the size of $b_{\max}$. Hence, it may well be possible to use a value of $n = 1$ and use a higher value of $b_{\max}$, which then means the proofs given in Section 6.1 of this metric's properties hold. In terms of mean throughput ($\max(\bar{b})$) the proposed metric had excellent performance compared to all other metrics, for both the constrained and unconstrained cases. As would be expected, this family of metrics also did well in terms of data transfers ($\max(bt)$). Significantly, when we examine for how many of the test cases this family of metrics had a data transfer within 10% of the maximum achieved for all metrics for that trace, this family of metrics did better still (particularly the $((b_{\max} - b)^2 + 1)t + h$ metric, which included a penalty for handovers), suggesting that the variation within this family of metrics is not particularly large.

In terms of percentage of time spent disconnected, the b_{target} metric performed best compared to all others excepting the $1/bt$ metric, which we have already seen

performs less well in terms of mean throughput. In particular, it should be noted that the constrained metrics almost always have a smaller disconnection time than the constrained ones, indicating that using proactive routing can also reduce disconnection times. This was not indicated by the results for the constrained case as given in Section 6.6.2, most probably due to the fact that the coverage map used did not cover a large number of roads in the city.

Finally, the last column of Table 6.5 shows that the proposed family of metrics performs best out of all the unconstrained metrics in terms of the length of time the route recommended takes to traverse, as compared to the shortest path route. These metrics generate routes that are within 10% (in terms of traversal time) of the shortest path route in all 20 test cases, whereas the other proactive metrics only did so for between 5 to 15 of the test cases.

Metric	$\min(h)$	$\max(\overline{b})$	$\overline{b} \geq b_{\text{target}} \pm 10\%$	$\max(bt)$	$\max(bt) \pm 10\%$	$\min(t_d\%)$	$\min(t) \pm 10\%$
$1/b$	5	1	6	1	2	1	15
$1/bt$	1	6	3	6	6	20	15
$1/bt + s$	20	1	0	1	1	1	5
$1/bt + t + h$	20	1	0	1	1	1	5
$(b_{\max} - b + 1) * t$	3	15	0	14	16	4	20
$((b_{\max} - b)^2 + 1) * t$	3	11	0	10	17	4	20
$((b_{\max} - b)^3 + 1) * t$	2	12	0	11	18	4	20
$((b_{\max} - b)^4 + 1) * t$	2	11	0	11	18	4	20
$((b_{\max} - b)^3 + 1) * t + h$	3	1	0	2	17	1	20
$(b_{\text{target}} - b + 1) * t$	2	6	3	6	6	10	20
SP $1/b$	5	1	10	1	1	1	20
SP $1/bt$	3	1	5	3	4	1	20
SP $1/bt + s$	9	1	4	1	1	1	20
SP $1/bt + t + h$	9	1	4	1	1	1	20
SP $(b_{\max} - b + 1) * t$	3	1	0	4	14	1	20
SP $((b_{\max} - b)^2 + 1) * t$	3	1	0	4	14	1	20
SP $((b_{\max} - b)^3 + 1) * t$	3	1	0	4	14	1	20
SP $((b_{\max} - b)^4 + 1) * t$	3	1	0	3	14	1	20
SP $((b_{\max} - b)^3 + 1) * t + h$	3	1	0	1	12	1	20
SP $(b_{\text{target}} - b + 1) * t$	3	1	3	3	5	1	20

Table 6.5: Number of test cases (out of 20) where each proactive metric performed best in a particular category (column). For each metric, the algorithm run without a constrained route, and again over the shortest geographical path (SP) route. Note that each column may add up to greater than 20, as multiple metrics may have performed equally well. Weightings for each term in the metrics have been omitted for clarity, but are the same as those given in Section 6.5.5.

6.7.5 Relation to Use Cases

Looking back to the use cases given in Section 5.1.2, the goal of maximising the amount of data transferred, as discussed in Section 6.6.1, has been satisfied. As regards minimising handovers and disconnection times, the results show that the proactive algorithms *increase* these quantities in many cases (although they do decrease them in some cases). There are three reasons for this: firstly, the usage of proactive algorithms increases the number of networks that are determined as viable to connect to, and thus the number of inter-network handovers increases. Such increases could be negated by increasing the penalty incurred by traversing a handover edge. Secondly, the reactive algorithm that was used is quite conservative about which networks it attempts to connect to. If conditions such as preferring an upward trend in RSS are relaxed, then a greater number of handovers would take place. However, this would result in a greater number of connections to networks that turn out to not be available for long enough to yield any benefit. Thirdly, if the lowest possible time spent disconnected is required, it would seem most sensible to remain connected to the cellular network, and only handover in areas where cellular coverage is non-existent. A proactive approach will perform better in these coverage blackspots as it will select the most useful networks, but if the cellular network is always available then there is no need to optimise which networks are chosen (and hence no reason to use any sort of proactive algorithm). Where a proactive approach is useful is for achieving low disconnection times whilst *simultaneously* maintaining a throughput that is above a particular target. Here, the client is forced to perform handovers as there is no ubiquitous network that can support the target throughput. It has been demonstrated that a proactive metric can be used for unconstrained routing to achieve this given target throughput, though further work is needed to improve performance.

6.8 General Applicability

It is also worth emphasising that the usage of of a proactive scheme does not preclude falling back to other schemes. Indeed, it is very likely that there will be areas for which coverage maps are not available, or where there is no known coverage, for which reactive mechanisms should be utilised. Similarly, opportunistic networking may also have a rôle to play, both for supplying connectivity via delay tolerant networking, (e.g. DieselNet [21]) or for simply allowing cheaper data transfer than might be possible over, say, an existing cellular network.

Also, it is important to point out that this work has made the assumption that a particular TCP, UDP or similar stream may only be conducted over a single physical network interface at any one time. This work still has relevance even when multiple interfaces can be used concurrently for the same stream, as it is still important to know when a particular network should be utilised in preference to another of the same technology. Additionally, handovers will still be relevant because

the aggregate goodput of all network interfaces will (evidently) change. Therefore, even if multiple physical interfaces are "bonded" together, routing algorithms that are QoS-aware will still be an important tool. Finally, it should be noted that many providers assign private IP addresses behind a single public address, and use Network Address Translation that is stateful. This tends to mean that each TCP connection will need to be confined to a single provider's connection, rather than striped over multiple ones. However, this of course does not prevent multiple TCP connections each of which is solely assigned to a particular physical interface.

6.9 Chapter Summary

This Chapter presented a novel method for generating routes for vehicles that not only takes into account distance and time, but also network connectivity and inter-network handovers. This QoS-aware routing utilises the multi-planar graph structure proposed in Chapter 5, which in turns depends on the coverage maps proposed in Chapter 4. This proactive approach to handovers was evaluated using six different metrics using real data over 25 geographically constrained routes and compared to a realistic reactive algorithm. The results showed that the proactive algorithm improves mean throughput in above 75% of test cases, frequently increasing mean throughputs by at least 1 Mbit/s. These results could be further improved if more information concerning the prevailing traffic speed on each road were known, information which is commercially available.

The proactive algorithm was then applied to unconstrained routing. The cross-city routes suggested by 10 different metrics were examined, and compared to the network performance that would be experienced if the shortest path routes were taken instead (but the same metric used in constrained mode). It was found that the family of routing metrics proposed performed significantly better than the other proactive routing metrics. In addition, proactive metrics, when run in the unconstrained case, provided better mean throughput and data transfer sizes than the same metrics when constrained to the shortest path routes. Importantly, the proactive metrics did not increase the journey times between the start and end points of the test cases by more than 10% in all 20 test cases.

Thus, it has been demonstrated that the proactive algorithm achieves its goals of providing superior network performance over a given route than a reactive one. More importantly, it is able to solve the problem of routing that takes into account not only distance but also network performance, as outlined in Section 5.2.

7

Conclusions

THE conclusions of this dissertation are summarised in this Chapter, beginning with an overall summary. The research questions proposed in Chapter 1 are once more examined in the light of the results obtained, before moving on to examine the work's weaknesses and its wider applicability. The Chapter concludes by overviewing topics for further research, notably the requirement for a new framework for representing mobility, and the possibilities of using VSNs in cognitive radio deployments.

7.1 Summary

Transportation is vital to today's society, with huge economic importance. Whilst travel has many benefits, Chapter 1 outlined how transportation is also responsible for many deaths. As a result, more and more computing infrastructure is being deployed to make vehicles safer, cheaper, and more efficient. Meanwhile, connectivity is also increasing our ability to access more information at greater speeds. Connectivity whilst on the move, particularly whilst in vehicles, has come of age. GSM subscribers worldwide now exceed 3 billion[1], and the take up of mobile data offerings is also increasing. Applications of connectivity in vehicles range from uploading telematics data and downloading map updates, to downloading entertainment and advertisements [39].

Connectivity is near-ubiquitous, but it is inherently heterogeneous in nature. Chapter 2 overviewed the variety of wireless networking standards available for use with vehicles, and how cities are hosts to thousands of overlapping wireless deployments. The difficulties inherent in communicating with vehicles were also identified, particularly those due to the need for the network in use to be changed frequently.

Handovers between different networks cause disruption, both in the form of a period of disconnection of multiple seconds, and the adjustment that must be made by transport protocols such as TCP. A variety of intelligent handover algorithms have been proposed that aimed to decrease this disruption, many of them requiring a knowledge of each wireless network's coverage. However, no viable implementations of such coverage maps existed.

[1] http://www.gsmworld.com/about/index.shtml

Chapter 3 of this dissertation introduced the concept of Sentient Transportation, where vehicles are aware of their environment by means of onboard sensors. Many such vehicles form a vehicular sensor network. Using the Sentient Van platform, a large corpus of data was collected over three years, including 10 million UMTS and IEEE 802.11b/g received signal strength readings. In addition, experiments were performed on factors influencing how these off-the-shelf technologies perform in the vehicular environment.

Novel algorithms for processing these large quantities of signal strength data into coverage maps that were compact and yet accurate were proposed in Chapter 4. These enable networks to be intelligently selected based on their coverage.

Finally, Chapters 5 and 6 detailed how such coverage maps could be used to construct multi-planar graphs. These contained details of all available wireless networks. This novel approach allowed efficient solutions to be found to the problem of what networks should be connected to over a journey.

7.2 Research Questions Addressed

In Section 1.4 three wide-ranging research questions were proposed that were to be addressed in this dissertation. Specifically:

- What performance can be expected from wireless technologies in vehicular environments?
- How can an accurate, compact, model of the wireless environment be constructed?
- How can such models be used, particularly for optimising communications systems?

Each of these will now be examined in turn.

7.2.1 Performance of Wireless Technologies for Vehicles

In order to investigate off-the-shelf communications technologies for vehicles, a suitable platform was required. As described in Section 3.1, the Sentient Van was constructed with a wide variety of onboard sensors and communications equipment. Moreover, it has been used for normal journeys for a period of over three years. This has enabled a corpus of over 56 million readings to be obtained, 10 million of which have concerned UMTS and IEEE 802.11b/g networks. This data set provided a unique insight into how network signal strengths vary with location.

In using networking technologies outdoors, there is normally an expectation that meteorological parameters will affect performance. In order to test this, 2.6 million

UMTS and 802.11b RSS readings were logged over several months (Section 3.2). These data were interpolated with half-hourly readings from a local weather station. No correlations between RSS and any of the parameters recorded was observed. Neither were any significant temporal variations seen. This suggests that the coverage and expected air interface throughput of these networks is fixed at a given location.

As observed in Section 3.4, much work has been carried out concerning IEEE 802.11b/g throughputs, but very little on 802.11a. However, the latter is what the 802.11p standard for vehicular communication is based on. Hence, experiments were performed to investigate this technology. A controlled, indoor environment was used to examine how access point positioning and beacon interval affect throughput. These factors impact the deployment cost and the seamlessness of mobility respectively, and hence are crucial in large network deployments.

Another observation made was that 802.11a had not been tested at the *low* speeds typical of congested city roads. Section 3.5 described outdoor experiments that measured the variation in throughput linked to speed. The results showed that very low speeds cause a larger variation in throughput, due to traversing regions of deep fade for longer. This observation is in contrast to previous work concerning *high* speeds, where no adverse effects on throughput were reported.

7.2.2 Constructing a Model of the Wireless Environment

The very large data set obtained using the Sentient Van brought the issue of how to process such sensor data to the fore. As described in Section 4.1, previous approaches attempted to process data that was distributed over all space. This dissertation has focused on the constrained problem domain of vehicles, where data points are located on roads. The advantage of this is that only two dimensions need to be considered, namely the position of each point in terms of the length of the road it is located on, and its sensor value. The problem is therefore transformed into one of curve fitting and segmentation.

Chapter 4 described existing approaches to representing noisy data. A key aim was to generate a representation that could be easily queried, and was compact. Compression of the data (e.g. by wavelets) does not produce a form that can easily be used for routing. Hence, algorithms were adapted from the field of 2-D shape simplification to generate an *extents* representation of the data.

The evaluation of these coverage mapping algorithms was in two parts. Firstly, synthetic data was generated that followed a known distribution, and used as input to the algorithms (Section 4.7). This allowed their output to be compared against a known, true value. In addition, the compression ratios achieved (number of input points versus number of output points) were calculated. These tests provided confidence that the algorithms were fit for purpose.

Secondly, real traces were taken from the Sentient Van and the RSS values seen on the journeys compared to the predictions made by the coverage maps generated by the algorithms (Section 4.8). This is in contrast to previous work on coverage mapping in that it not only uses a large corpus of real input data (as opposed to simulation), but then evaluates the accuracy with traces taken in a city environment by a vehicle. The results obtained show that the 90% confidence interval of error in prediction was as low as 12 dBm for UMTS RSS and 10 dBm for IEEE 802.11b/g. Given that in Section 3.2 it was found that at a given location the standard deviation of UMTS RSS values was 3 dBm, and 3.5 dBm for IEEE 802.11b/g, the extra error in prediction due to the coverage mapping algorithms can be seen to be low.

In terms of how compact a representation was generated, the number of extents per metre of road varied significantly between algorithms (Section 4.8.2). Overall, one extent per (at least) 30 metres of road was required. This implies that a standard access point's coverage of 400 metres in length can be represented in twelve pairs of numbers. Whilst this figure could be further improved, it is low enough that the coverage database for an entire city will require only a modicum of storage.

As an illustration of the more general applicability of the algorithms proposed, other types of data were also processed. The maps of speed, ambient noise, and carbon dioxide concentration given in Section 4.12 showed visually how the domain of such algorithms is not solely confined to wireless RSS data.

7.2.3 Optimising Communications Systems for Vehicles

A key goal of the coverage mapping algorithms proposed in Chapter 4 was to provide a representation that could be easily converted into graph form. Chapter 5 detailed how a coverage map is converted to a multi-planar graph, a representation that has not been proposed in any previous work. The inclusion of coverage data for *all* available wireless networks, and the explicit specification of the costs of handovers between them has not until now been researched. By incorporating such handover penalties network selection decisions are more informed: the benefits of connecting to another network must be offset against the costs of the handover. Moreover, such coverage information enables applications to be better prepared for the disruption of handovers, such as by changing TCP window sizes.

The multi-planar graph structure that was initially proposed was subjected to a complexity analysis (Section 5.4.1). It was found that this is related to the cube of the number of different networks, as well as the mean number of extents per road. In order to decrease the complexity, sparse and zero-coverage planes were proposed (Section 5.4.2). As a result, a significant reduction in graph complexity was achieved (four orders of magnitude for an entire city's coverage map). Whilst the size of the graph is still large, these reductions made its inclusion within a portable navigation device more plausible.

In order to use the multi-planar graph to enhance network selection and handover timing, the use of shortest path routing was proposed. Chapter 6 provided an overview of how QoS-aware routing related to multi-objective programming, and showed that finding *optimal* solutions to the problem was $\mathcal{NP}$-hard. However, a family of routing metrics was proposed, with a theoretical underpinning, that were used to find *efficient* solutions to the network selection problem.

The system was again evaluated using traces from the Sentient Van. This was another distinction from previous work on handovers, which has tended to use either unrealistic movement models, or simulated wireless technology performance. The proactive routing metrics were compared to a non-trivial reactive algorithm over each of the 25 input traces (Section 6.5.2). In over 75% of the test cases, the proactive approach obtained better network QoS in terms of mean throughput than the reactive approach.

A natural extension of optimising QoS for vehicles is to not only perform network selection on a given (constrained) route, but to allow route selection to take QoS into account. This was addressed in Section 6.7, where 10 different routing metrics were used to find QoS-aware routes between 20 pairs of random points. The results were compared to the QoS that would have been achieved (according to the coverage map) had the shortest path route been used instead. The results showed that mean throughputs were greater using proactive, unconstrained routing in 75% of the cases tested, with even better results when quantity of data transferred was considered. These routes were also no longer than 10% extra of the corresponding shortest path route, suggesting that the metrics were correctly formulated to also take time into account. Therefore, the system provides a proven method of optimising vehicular connectivity. Moreover, provided a coverage map exists, the use of the system requires *no* augmentation of existing infrastructure, and has no minimum penetration requirement.

7.3 Overall Evaluation

Whilst the above has examined how this dissertation has answered the original research questions that were posed, it is important to examine the deficiencies and wider applicability of the systems proposed. This Section fulfils these goals.

7.3.1 Weaknesses

7.3.1.1 Reliance on a Vehicular Sensor Network

Coverage maps rely on the ability to collect large quantities of RSS data. Clearly, there is a cost associated with obtaining such data, and the question to be asked is whether it is economically feasible to do so. Fortunately, various options are possible.

Firstly, many delivery firms such as UPS[2] already have telematics platforms installed on their vehicles for reporting vehicle performance data and location back to base. Hence, fleets are already available for data collection. Moreover, taxi and delivery fleets visit many different locations each week, thus ensuring that much of the road network is visited.

Secondly, firms such as Tele Atlas and Navteq already use probe vehicles when creating digital road maps. These vehicles collect data concerning turn restrictions and lane information, as well as images, which are then incorporated into the companies' mapping products. Thus, augmenting these vehicles to log RSS values would be simple.

Hence, collecting the required quantity of RSS data is likely to be relatively easy, particularly given that the coverage maps generated would be useful to the entities performing the data collection, thus providing an incentive to do so.

7.3.1.2 Inexact Relation of Throughput to RSS

Section 2.3.3 described how many wireless technologies use adaptive modulation and coding schemes in order to trade-off robustness with throughput. This means that the possible throughput depends on the SNR, which in turn depends on the received signal strength. The coverage maps produced in Chapter 4 are of RSS values, and these are converted to throughputs using the tables in Section 6.4.2.

Clearly RSS is not the sole factor that influences throughput, as illustrated in the experiments concerning beacon interval in Section 3.4.4. Crucially, the number of users that are simultaneously utilising a particular access point or base station will change the throughput that can be achieved. This is particularly true for a distributed medium contention-based technology such as IEEE 802.11x.

Two solutions exist to this problem. The first involves the infrastructure reporting its utilisation level, either to all clients, or to a central repository which then distributes the information. Whilst this approach is possible, it would require enormous changes to deployed infrastructure. The second, more plausible, option is to record measures of throughput rather than RSS. These readings could then be used as input to the coverage mapping algorithms. The main reason that this was not

[2] See `http://www.cio.com/article/355913/` for details.

done for this work is that the number of WiFi networks that could be *legally* used was very few, and hence obtaining a coverage map of throughput would have been difficult. In any case, the type of input data is irrelevant as regards the proposed algorithms for producing the coverage maps and constructing the multi-planar graph representation. Hence, this is not regarded as a significant weakness.

7.3.1.3 Constrained to Paths

In constructing coverage maps the simplifying assumption was made that they would only cover well-defined paths, such as the road network. Whilst this works well for vehicles, it is not applicable to general pedestrian movement. However, given that vehicles are used for the majority of travel that occurs at high speeds and over long distances, this is regarded as the most important application domain for optimising connectivity. Of course, the same system could be used by pedestrians when walking along roads or footpaths, and thus it is still applicable to many (but not all) pedestrian journeys.

7.3.1.4 Multi-Planar Graph Complexity

Despite the complexity reduction techniques proposed in Section 5.4.2, the size of the multi-planar graph as compared to the road network is large. Table 5.3 shows how for a road network of 5,731 edges increased to having 76,430 edges in the multi-planar graph. This expansion factor will increase as more wireless networks are added.

However, as noted in Section 4.9.2, the set of wireless networks included in the coverage map is unlikely to include *all* deployments, but instead those that a user has a subscription with. Hence, the complexity of the graph will be somewhat reduced (though given the number of nodes belonging to some community WiFi networks, there will still be significant choice[3]).

Finally, whilst at present personal navigation services are run on relatively low-resourced devices, this is changing. The next generation of such devices will include imagery of city streets, implying that storage is not particularly constrained. In addition, many more services will move online, with the devices used to access online route planning portals, rather than necessarily holding the coverage map locally.

[3]See, for example, Fon's AP density map, `http://www.bt.com/static/wa/wifi/pages/findhotspots.html`.

7.3.2 Wider Applicability

7.3.2.1 Applicability to Other Forms of Transport

Whilst the work presented in this dissertation has predominantly used the Sentient Van, it is important to note that this work is not restricted to motorised road vehicles (bikes being one possible use case), but is also applicable to other forms of transport. In particular, there is a great deal of activity centred around providing broadband access for trains. Two main forms of this exist, the first using satellite connectivity, the second utilising many different cellular networks (and in the future WiMax) in concert. The latter method currently uses a reactive mechanism to choose the current "best" network, but could benefit significantly from utilising the proactive approach proposed in this dissertation.

7.3.2.2 Applicability to Other Data Types

As seen in Section 4.12, coverage mapping is not confined solely to RSS data. Other data can also be mapped, such as the traffic speeds, concentrations of air pollutants, or the steepness of individual roads. In this respect, the algorithms developed are suitable for any type of mobile sensing that runs along known paths. By using vehicular sensor networks and the proposed algorithms to generate accurate maps of data types such as pollutants, a better understanding can be gained of the small scale effects such as pollutant hotspots, rather than city-wide models. Such data will be key in enabling schemes such as routing traffic away from polluted areas, or allowing individuals to decide where they might live based on the pollutant concentrations on particular streets.

Moreover, multi-criteria routing can also be applied to other data types. Whether this is a simple network graph this is augmented with a quantity to be minimised, such as level of air pollution along a route, or a more complex conflicting criteria problem, the methods for converting a coverage map into graph form given in Chapter 5 remain applicable.

Other problems may map well onto the multi-planar graph paradigm. One example is deciding when to use the electric-drivetrain (instead of a fossil fuel combustion engine) of a hybrid vehicle, where switching between the two modes might incur a start-up "penalty". This is one possible area of future research.

7.3.2.3 Sensor Data Discard

Another application of coverage maps is to give sensor platforms hints on how much data to store and what to discard. For example, if it is known that the vehicle will not experience high throughput, low cost coverage for the next hour, there is no need to store video data on the current traffic state that is useless after five minutes.

Conversely, if there is to be a high throughput connection within one minute the data should be stored for upload. Similar situations arise with the frequency of reporting of data, such as engine performance values to the manufacturer, where it is of greater utility if there are more frequent updates, but this is traded off with the cost of uploading such data.

7.4 Further work

A variety of questions and topics for further research have arisen from the work presented in this dissertation. These are divided into specific queries concerning the work presented here, and more general, related areas.

7.4.1 Specific Open Questions

Update frequency of coverage maps Section 4.10 outlined the need for coverage maps to obtain frequent enough updates in order to remain accurate. Whilst the data collected over the past 3 years does show some churn in deployed networks, further analysis is necessary to derive how frequently data collection needs to be carried out.

Minimum data set required The coverage maps generated in Chapter 4 utilised data collected over 1.5 years. The only requirement for generating extents was a qualitative one of a minimum of 50 points per road. Analysis was not carried out to determine what the minimum number of points necessary for a given prediction accuracy is, but would be useful to determine the number of probe vehicles required.

Weighting recent values more heavily At present all data points are given equal significance by the coverage mapping algorithms. Whilst points older than a certain age could be discarded, a better method would be to weight points according to age. This would mean maps would be biased towards newer readings, whilst still making use of old data to guard against transient readings.

Target throughput metric The metric for finding a route above a given target throughput described in Section 6.7.1 did not perform well. Whilst this was in part due to the relative sparseness of the coverage map, modifying the metric to have a higher coefficient of n might produce better results. Further work on adjusting the metrics' parameters would also be of interest.

Partial loop unrolling Section 6.3.5 proposed partial loop unrolling as a solution to the problem of how to allow pauses on a journey that would enable a particular data transfer. This technique has not been implemented or evaluated for this dissertation, but appears a viable approach that could be progressed further.

Improved awareness of prevailing road speeds The performance of the proactive algorithms is in part dependent on the accuracy of the database of road speeds used to calculate how long a vehicle spends on each edge of the graph (Section 6.5.3). If a commercial database were purchased, this would enable more accurate (including time-of-day dependent) routing calculations.

Temporally dependent maps Whilst RSS is unaffected by meteorology or time of day (Section 3.2), factors such as traffic speeds and number of wireless network users are. Hence, multiple coverage maps could be created, e.g. one for rush hour and another for other times of day.

7.4.2 The Need for a New Framework for Mobility

The Open Systems Interconnection (OSI) model [257] was conceived as an abstraction of how the different layers of a network should operate. Specifically, each layer should provide particular service to those above it; a technology supplying functionality at one layer should be able to be replaced by another at the same layer without the other layers being aware of it; each layer would have a notion of end-to-end connectivity that abstracted away from the layers below it.

Today, mobility and vertical handovers have meant that the OSI model is no longer suitable for modelling modern networking. The peripheral (wireless) networks have significantly different throughputs and error properties when compared to the fibre optic networks used in the core, despite being at the same notional OSI layer. End-to-end connectivity is lost with proxies/agents that cope with mobile nodes changing their points of attachment. Applications perform best when they are aware of what the physical layer is capable of, and when its characteristics will change (e.g. changing the bit rate used for a multimedia transmission when the connection changes from WiFi to UMTS).

In order to carry out research into how proactive handover schemes can be deployed, a new model has been proposed (by the author, in conjunction with others), to reflect how mobility has affected networking. The Y-Comm framework [47, 158] uses different subsystems for core and peripheral networks, with cross-layer communication built in. Further research is necessary to bring this proposal to implementation.

7.4.3 The Rôle of VSNs in Cognitive Radio

Another possible application of coverage maps is in the area of spectrum sensing and cognitive radio. Software Defined Radio consists of hardware that is capable of using any frequency within a very wide band, and moreover can change its transmission modulation scheme and media access control layer to that of any of a variety of technologies. The idea behind this scheme is that one device will be capable of connecting to a wide variety of networks whilst only having one physical network interface.

Cognitive Radio (CR), meanwhile, is proposed as a mechanism to use radio spectrum more efficiently. When a CR wishes to transmit, it scans a wide frequency band to find spectrum where there are currently no transmissions taking place. It then uses this band for transmission, using the appropriate modulation (etc.) scheme on its software defined radio [197, 7].

CR has attained prominence in regard to how the spectrum freed by the move of analogue TV broadcasts to digital transmission will be used. One possibility is that the spectrum will be licensed out to *primary* users (who pay for it), but will also permit the use of CRs as *secondary* users [256]. Because there will be geographical areas and time periods when the primary user of a band is not using the spectrum, secondary users may then transmit in it.

One of the problems of CR is the time taken to scan a wide frequency band can be several minutes long. If a CR were deployed on a vehicle, this might mean that its location had changed significantly between starting and ending the scan. Hence, the frequency chosen might no longer be unoccupied. One proposed solution to this is the use of *beacons*, which inform CRs in the vicinity of what frequencies are in use [208]. The beacon transmitters would hold databases of what frequencies are in use in what areas, the databases being populated by sensing infrastructure.

Coverage mapping builds on the idea of detecting available networks, i.e. sensing what spectrum *is in use*. If, however, it were employed for the inverse task of sensing and mapping what spectrum was *not* in use, the resulting maps could act as a guide for CRs. Moreover, because these maps could include time-dependent (as well as the usual location-dependent) information, the bands that a CR would need to scan for unused frequencies would be far fewer. Thus, scanning times could be significantly reduced, increasing the utility of CRs whilst on the move.

Finally, not only could vehicular sensor networks build up maps of unused spectrum, but they could also be used for enforcement. At present, if an entity transmits on a primary user's spectrum (a "pirate" station), tracking down the source of the transmission is difficult. This will only increase if secondary users are specifically *permitted* to use primary bands, subject to them being unoccupied. Using a vehicular sensor network, spectrum sensing could be used to ascertain the location of any non-conformant user, thus easing enforcement activities. Such enforcement will be particularly important if spectrum is dynamically auctioned [240].

A

Approaches to Curve Representation

A.1 General Methods of Curve Fitting

There are a great number of different approaches to fitting a curve to a set of noisy data. The well-known least squares approach to fitting a data set normally[1] attempts to fit a linear function to the data whilst minimising the mean squared error of the sample points as compared to the approximating curve. Given the RSS curve cannot be assumed to be linear, such an approach is infeasible.

Spline fitting is a more flexible, yet complex approach. A spline consists of a set of polynomials that are constrained by knot points. The knot points may be distributed evenly over the range that the spline is to represent (uniform splines) or unevenly (non-uniform splines), and ensure that the end of each polynomial is co-located with the start of the next. In addition, smoothness constraints may be enforced, in order that higher order differentials of the two polynomials meeting at each knot point are equal. Using splines to fit noisy data was originally proposed by Craven and Wabha [45], using shifted Bernoulli polynomials with one knot, in addition to $m + 1$ polynomials of degree m. Unfortunately such methods require the function that is to be represented to be smooth, which we have already seen is not necessarily the case. Additionally, the output is likely to involve a significant number of polynomials, which would make ascertaining the value of the function at a given co-ordinate a relatively complex operation.

A.2 Line Simplification

A standard approach to the problem of line simplification is the Douglas-Peucker algorithm [66], an example of which is shown in Figure A.1, which is used widely in areas such as coastline simplification for generating smaller scale maps. The algorithm takes as its input an array of points, $A = [a_0..a_n]$. It initially creates an approximation line between a_0 and a_n (shown dashed in Figure A.1), and measures the perpendicular distance of all the intervening points from that line. If any of these distances is greater than a given threshold, ϵ, then the portion of the array that the approximation line covers is partitioned into two. Hence, in the second

[1] Non-linear least squares is also possible, but does not have a closed solution, and must be solved by iteration.

iteration of the algorithm, two lines will be created, one from a_0 to $a_{n/2}$, and another from $a_{n/2}$ to a_n. The process is then repeated on each of these lines, until all the original points are within ϵ of their corresponding approximation line.

Other, similar, approaches to line simplification have been devised, and are well surveyed in [133] and [102].

These algorithms assume that all points of the input data are equally significant, i.e. that there are no spurious outliers due to noise. In addition, they assume a single input value at each location, rather than several co-located values. This is not the case for RSS values shown in Figure 4.1. Hence, a more complex technique is necessary.

A.3 Curve Decomposition

Another important technique for representing digital curves is to decompose them into their constituent frequencies. This can be done by repeated smoothing, such that after each stage a different frequency of wave is visible. This component is then subtracted, and the process repeated, until the remaining wave is ascertained to be noise [78]. In a similar fashion, wavelets, rather than oscillatory curves, can be used [93]. Wavelets have the advantage that they are of finite width (rather than infinite, as it the case with oscillatory curves). In either case the most significant subcomponents are stored. These techniques are (again) useful when every aspect of the original curve is of equal significance, but do not lend themselves to direct application to noisy RSS data (though recent work has been more applicable in this regard, e.g. see [28]). In addition, their output (frequency components or wavelets) are not easily queried by a database, where all the components would need to be reassembled in order to ascertain the value at a particular point.

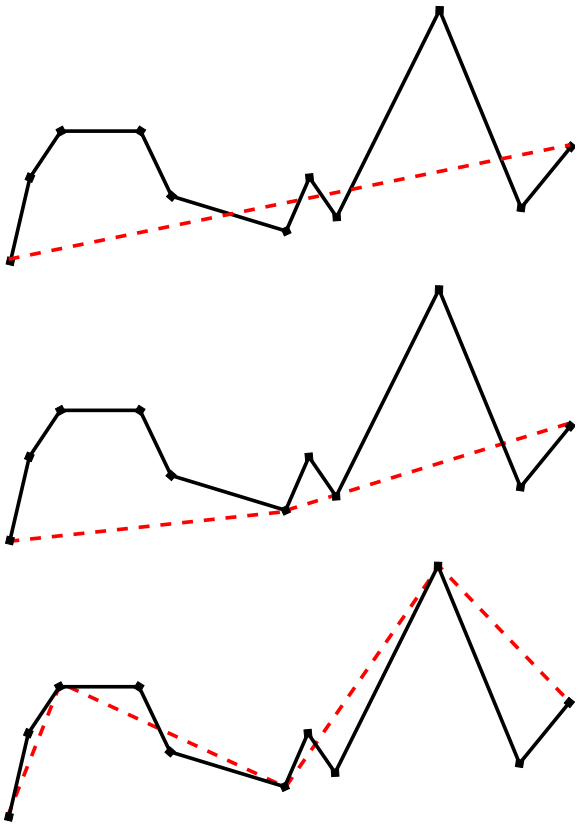


Figure A.1: Douglas-Peucker line simplification: at each step, the maximum distance of each red approximation segment to the points it is representing is calculated. If this is greater than a specified threshold, the number of points that segment is to approximate is divided by two. This is repeated until the representation is good enough that all red segments are less than the specified tolerance from the points they represent on the original curve.

References

[1] 3GPP. *Universal Mobile Telecommunications System (UMTS); UTRA high speed downlink packet access (3GPP TR 25.950 version 4.0.0 Release 4)*. ETSI, March 2001. 56

[2] 3GPP. *3GPP Technical Specification 25.213 version 7.5.0 Release 7: Universal Mobile Telecommunications System (UMTS) Spreading and Modulation (FDD)*. ETSI, June 2008. 53

[3] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-management in chaotic wireless deployments: Extended version. *Wireless Networks*, 13(6):737–755, December 2007. 28, 119

[4] AMI-C. AMI-C use cases. Technical Report 1001, AMI-C, January 2003. 30

[5] Applied Generics Ltd. Rodin24 road traffic monitor (GSM edition). Technical report, Applied Generics Ltd., 2004. 40

[6] ARINC. Navstar GPS space segment/navigation user interfaces. Technical Report IS-GPS-200, Revision D, Navstar Joint Program Office, December 2004. 51, 64

[7] Steven Ashley. Cognitive radio. *Scientific American*, pages 66–73, March 2006. 237

[8] S. Aust, N. Fikouras, D. Protel, C. Gorg, and C. Pampu. Policy based mobile IP handoff decision (POLIMAND). Technical Report draft-iponair-dna-polimand-02.txt, Work in Progress, IETF, February 2005. 63

[9] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM*, volume 2, pages 775–784, 2000. 64

[10] A. Baig, M. Hassan, and L. Libman. Prediction-based recovery from link outages in on-board mobile communication networks. In *Proc. IEEE GLOBECOM*, November-December 2004. 121

[11] Mary G. Baker, Xinhua Zhao, Stuart Cheshire, and Jonathan Stone. Supporting mobility in MosquitoNet. In *Proc. USENIX*, pages 120–127, January 1996. 59, 63

[12] Joshua Bardwell. You believe you understand what you think I said. Technical report, Connect802 Corporation, 2004. 45

[13] R. Battiti, A. Villani, and T. Le Nhat. Neural network models for intelligent networks: deriving the location from signal patterns. In *Proc. AINS*, 2002. 66

[14] Roberto Battiti, Mauro Brunato, and Alessandro Villani. Statistical learning theory for location fingerprinting in wireless LANs. Technical Report DIT-02-0086, Department of Information and Communication Technology, University of Trento, Italy, October 2002. 66

[15] M. Bazaraa and C. Shetty. *Nonlinear Programming, Theory and Algorithms*. Wiley, 1979. 187

[16] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958. 182

[17] P. Bergamo, D. Maniezzo, K. Yao, M. Cesana, G. Pau, M. Gerla, and D. Whiteman. IEEE 802.11 wireless network under aggressive mobility scenarios. In *Proc. International Telemetry Conference*, October 2003. 108

[18] Richard Bishop. *Intelligent Vehicle Technology and Trends*. Artech House, 685 Canton Street, Norwood, MA, USA, 2005. 37

[19] J. Blum, A. Eskandarian, and L Hoffman. Challenges of intervehicle ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):347–351, December 2004. 47

[20] John Brownfield, Andy Graham, Helen Eveleigh, Faber Maunsell, Heather Ward, Sandy Robertson, and Richard Allsop. Congestion and accident risk. Technical Report Department for Transport Road Safety Report Number 44, Centre for Transport Studies, University College London, November 2003. 26

[21] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, April 2006. 48, 224

[22] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *Proc. ACM WSW*, October 2006. 39

[23] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden. Data management in the CarTel mobile sensor computing system. In *Proc. ACM SIGMOD*, pages 730–732, 2006. 41

[24] Vladimir Bychkovsky, Bret Hull, Allen K. Miu, Hari Balakrishnan, and Samuel Madden. A measurement study of vehicular internet access using in situ Wi-Fi networks. In *Proc. ACM MobiCom*, Los Angeles, CA, September 2006. 28, 40, 77, 109, 119

[25] Simon Byers and Dave Kormann. 802.11b access point mapping. *Communications of the ACM*, 46(5):41–46, 2003. 124

[26] Joseph Camp and Edward Knightly. Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation. In *Proc. ACM MobiCom*, pages 315–326, September 2008. 115

[27] Srdjan Čapkun and Jean-Pierre Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, February 2006. 49

[28] Daniel Castaño and Angela Kunoth. Adaptive fitting of scattered data by spline-wavelets. In Albert Cohen, Jean-Louis Merrien, and Larry L. Schumaker, editors, *Curves and Surfaces*, pages 65–78. Vanderbilt University Press, Nashville, TN, USA, 2003. 240

[29] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. A probabilistic room location service for wireless networked environments. In *Proc. UbiComp*, volume 2201 of *LNCS*, pages 18–34, 2001. 64

[30] R. Chakravorty, J. Cartwright, and I. Pratt. Practical experience with TCP over GPRS. In *Proc. IEEE GLOBECOMM*, pages 1678–1682, November 2002. 59

[31] James C. Chen. Measured performance of 5-GHz 802.11a wireless LAN systems. Technical report, Atheros Communications, Inc., August 2001. 58

[32] Lin Cheng, Benjamin E. Henty, Daniel D. Stancil, Fan Bai, and Priyantha Mudalige. Properties and applications of the suburban vehicle-to-vehicle propagation channel at 5.9 GHz. In *Proc. ICEAA*, pages 121–124, September 2007. 109

[33] Y. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale Wi-Fi localization. In *Proc. ACM MobiSys*, pages 233–245, 2005. 77, 123

[34] Boris V. Cherkassky and Andrew V. Goldberg. Negative-cycle detection algorithms. *Mathematical Programming*, 85(2):277–311, June 1999. 185

[35] Mashrur A. Chowdhury and Adel Sadek. *Fundamentals of Intelligent Transportation Systems Planning*. Artech House, 2003. 37

[36] Jared L. Cohon. *Multiobjective Programming and Planning*. Academic Press, 1978. 186

[37] David N. Cottingham, Alastair R. Beresford, and Robert K. Harle. A survey of technologies for the implementation of national-scale road user charging. *Transport Reviews*, 27(4):499–523, July 2007. 23

[38] David N. Cottingham, J. J. Davies, and A. R. Beresford. Congestion-aware vehicular traffic routing using WiFi hotspots. In *Proceedings of Communications Innovation Institute Workshop*, pages 4–6. Cambridge-MIT Institute, April 2005. 24

[39] David N. Cottingham and Jonathan J. Davies. A vision for wireless access on the road network. In *Proc. 4th International Workshop on Intelligent Transportation*, pages 25–30, March 2007. 23, 227

[40] David N. Cottingham, Jonathan J. Davies, and Brian D. Jones. A research platform for sentient transport. *IEEE Pervasive Computing*, 5(4):63–64, Oct–Dec 2006. 22, 23

[41] David N. Cottingham and Robert K. Harle. Constructing accurate, space-efficient, wireless coverage maps for vehicular contexts. In *Proc. ICST WICON*, November 2008. 23

[42] David N. Cottingham and Robert K. Harle. Handover-optimised routing over multi-planar graphs for vehicles. In *In progress.*, 2009. 23

[43] David N. Cottingham and P. Vidales. Is latency the real enemy in next generation networks? In *Proc. ICST CoNWiN*, July 2005. 23, 60, 199

[44] David N. Cottingham, Ian J. Wassell, and Robert K. Harle. Performance of IEEE 802.11a in vehicular contexts. In *Proc. IEEE VTC*, pages 854–858, April 2007. 23

[45] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, December 1978. 239

[46] D. Crosby, V. Abhayawardhana, I. Wassell, M. Brown, and M. Sellars. Time variability of the foliated fixed wireless access channel at 3.5 GHz. In *Proc. IEEE VTC*, May 2005. 78

[47] Jon Crowcroft, David Cottingham, Glenford Mapp, and Fatema Shaikh. Y-Comm: A global architecture for heterogeneous networking. In *Proc. 3rd Annual International Wireless Internet Conference (WICON)*, October 2007. Invited paper. 24, 236

[48] John Current and Michael Marsh. Multiobjective transportation network design and routing problems: Taxonomy and annotation. *European Journal of Operational Research*, 65(1):4–19, February 1993. 189

[49] John Current and HoKey Min. Multiobjective design of transporation networks: Taxonomy and annotation. *European Journal of Operational Research*, 26(2):187–201, August 1986. 189

[50] Piotr Czyżak and Andrzej Jaszkiewicz. Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998. 188

[51] Layer D. H. Digital radio takes to the road. *IEEE Spectrum*, 38(7):40–46, Jul 2001. 51

[52] Jonathan J. Davies, Alastair R. Beresford, and Andy Hopper. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5(4):47–54, Oct–Dec 2006. 38

[53] Jonathan J. Davies, David N. Cottingham, and Brian D. Jones. A sensor platform for sentient transportation research. In *Proc. 1st European Conference on Smart Sensing and Context*, volume 4272 of *LNCS*, pages 226–229, October 2006. 22, 23

[54] Peter Day, Jianping Wu, and Neil Poulton. Beyond real time. *ITS International*, 12(6):55–56, November/December 2006. 40

[55] Antonio de la Oliva, Telemaco Melia, Albert Vidal, Carlos J. Bernardos, Ignacio Soto, and Albert Banchs. IEEE 802.21 enabled mobile terminals for optimized WLAN/3G handovers: a case study. *SIGMOBILE Mobile Computer Communications Review*, 11(2):29–40, 2007. 63

[56] Eric M. Delmelle, Peter A. Rogerson, Mohan R. Akella, Rajan Batta, Alan Blatt, and Glenn Wilson. A spatial model of received signal strength indicator values for automated collision notification technology. *Transport Research Part C*, 13:432–447, 2006. 126

[57] A. Demers, G. F. List, A. Wallace, E. E. Lee, and J. M. Wojtowicz. Probes as path seekers: A new paradigm. *Journal of the Transportation Research Board*, 1944:107–114, 2006. 40

[58] Jan Derksen, Robert Jansen, Markku Maijala, and Erik Westerberg. HSDPA performance and evolution. *Ericsson Review*, 84(3), 2006. 47, 59, 88

[59] DfT. Transport trends: 2007 edition. Technical report, UK Department for Transport, December 2007. 27

[60] Robert B. Dial. A model and algorithm for multicriteria route-mode choice. *Transportation Research*, 13B(4):311–316, December 1979. 187

[61] Almudena Diaz, Pedro Merino, Laura Panizo, and Alvaro M. Recio. Evaluating video streaming over GPRS/UMTS networks: A practical case. In *Proc. IEEE VTC*, pages 624–628, April 2007. 88

[62] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 182

[63] Marios D. Dikaiakos, Saif Iqbal, Tamer Nadeem, and Liviu Iftode. VITP: an information transfer protocol for vehicular computing. In *Proc. ACM VANET*, September 2005. 41

[64] Florian Doetzer, Florian Kohlmayer, Timo Kosch, and Markus Strassberger. Secure communication for intersection assistance. In *Proc. International Workshop on Intelligent Transportation*, March 2005. 49

[65] Sandor Dornbush and Anupam Joshi. StreetSmart traffic: Discovering and disseminating automobile congestion using VANET's. In *Proc. IEEE VTC*, pages 11–15, April 2007. 40

[66] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973. 239

[67] Christos Douligeris and Thanos Vasilakos. Mobile IP protocols. In Ivan Stojmenović, editor, *Handbook of Wireless Networks and Mobile Computing*, pages 529–552. John Wiley, 2002. 59

[68] D. Duchamp and N. Reynolds. Measured performance of a wireless LAN. In *Proc. 17th IEEE Conference on Local Computer Networks*, pages 494–499, September 1992. 88

[69] G. Durgin, T. S. Rappaport, and H. Xu. Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz. *IEEE Transactions on Communications*, 46(11):1484–1496, November 1998. 43

[70] Stephan Eichler. Performance evaluation of the IEEE 802.11p WAVE communication standard. In *Proc. IEEE WiVeC*, September 2007. 57, 109

[71] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G-S. Ahn, and A. T. Campbell. The BikeNet mobile sensing system for cyclist experience mapping. In *ACM SenSys*, pages 87–101, 2007. 41, 70

[72] Jakob Eriksson, Hari Balakrishnan, and Sam Madden. Cabernet: A content delivery network for moving vehicles. Technical Report MIT-CSAIL-TR-2008-003, MIT CSAIL, January 2008. 50, 199

[73] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proc. ACM MobiSys*, pages 29–39, June 2008. 41

[74] ETSI. Digital cellular telecommunications system (phase 2+); Universal Mobile Telecommunications System (UMTS); AT command set for User Equipment (UE). Technical Report TS 127.007 v. 8.3.0 R-8, ETSI/3GPP, April 2008. 122

[75] B. Evans, M. Werner, E. Lutz, M. Bousquet, G.E. Corazza, G. Maral, and R. Rumeau. Integration of satellite and terrestrial systems in future multimedia communications. *IEEE Wireless Communications*, 12(5):72–80, October 2005. 51, 59

[76] A. Festag, G. Noecker, M. Strassberger, A. Lbke, B. Bochow, M. Torrent-Moreno, S. Schnaufer, R. Eigner, C. Catrinescu, and J. Kunisch. 'NoW – network on wheels': Project objectives, technology and achievements. In *Proc. 5th International Workshop on Intelligent Transportation*, pages 211–216, March 2008. 48

[77] WAP Forum. Wireless profiled TCP. Technical Report WAP-225-TCP-20010331, WAP Forum, March 2001. 61

[78] David H. Foster. Automatic repeated-loess decomposition of data consisting of sums of oscillatory curves. *Statistics and Computing*, 12:339–351, 2002. 240

[79] F. Frederiksen, P. Mogensen, and J. Berg. Prediction of path loss in environments with high-raised buildings. In *Proc. IEEE VTC*, volume 2, pages 898–903, 2000. 43, 127

[80] Herbert Freeman. Computer processing of line-drawing images. *ACM Computer Surveys*, 6(1):57–97, 1974. 133

[81] R. H. Frenkiel, B.R. Badrinath, J. Borres, and R. D. Yates. The infostations challenge: balancing cost and ubiquity indelivering wireless data. *IEEE Personal Communications*, 7(2):66–71, April 2000. 50

[82] G. Gaertner and V. Cahill. Understanding link quality in 802.11 mobile ad hoc networks. *IEEE Internet Computing*, 8(1):55–60, January-February 2004. 88

[83] G. Gallus and P. W. Neurath. Improved computer chromosome analysis incorporating preprocessing and boundary analysis. *Physics in Medicine and Biology*, 15(3):435–445, July 1970. 131

[84] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP Completeness*. W. H. Freeman, 1979. 195

[85] R. Gass, J. Scott, and C. Diot. Measurements of in-motion 802.11 networking. In *Proc. IEEE WMCSA*, 2006. 108, 199

[86] Matthew Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2nd edition, April 2005. 57

[87] Jim Geier. *Wireless LANs: Implementing Interoperable Networks*. Macmillan Technical Publishing, 1st edition, 1999. 57

[88] Anastasios Giannoulis, Marco Fiore, and Edward W. Knightly. Supporting vehicular mobility in urban multi-hop wireless networks. In *Proc. ACM MobiSys*, June 2008. 62, 63

[89] Domenico Giustiniano Giuseppe Bianchi, Fabrizio Formisano. 802.11b/g link level measurements for an outdoor wireless campus network. In *Proc. IEEE WoWMoM*, pages 525–530, 2006. 85

[90] Tom Goff, James Moronski, D. S. Phatak, and Vipul Gupta. Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments. In *Proc. IEEE INFOCOM*, volume 3, pages 1537–1545, March 2000. 121

[91] Mohamed G. Gouda and Marco Schneider. Maximizable routing metrics. *IEEE/ACM Transactions on Networking*, 11(4):663–675, August 2003. 184, 194

[92] Martin Grade, Klaus Meier, Bernd Rech, and Andreas Lübke. Physical IEEE 802.11 – measurements in automotive environment. In *Proc. ITS*, June 2005. 77, 117

[93] Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2), Summer 1995. 240

[94] D. Greenwood and L. Hanzo. Characterisation of mobile radio channels. In Raymond Steele and Lajos Hanzo, editors, *Mobile Radio Communications*, pages 91–186. John Wiley, 2nd edition, 1999. 44

[95] E. Guizzo. Network of traffic spies built into cars in Atlanta. *IEEE Spectrum Online*, 2004. 38

[96] David Gunton. MILTRANS – millimetric transceivers for transport applications. In *Proc. IEE Conference on Automotive Electronics*, page 251, March 2006. 53

[97] Eva Gustafsson and Annika Jonsson. Always best connected. *IEEE Personal Communications*, 10(1):49–55, February 2003. 58

[98] David Hadaller, Srinivasan Keshav, Tim Brecht, and Shubham Agarwal. Vehicular opportunistic communication under the microscope. In *Proc. ACM MobiSys*, pages 206–219, June 2007. 63

[99] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. ACM MobiCom*, pages 70–84, 2004. 77, 122

[100] William W. Hardgrave and George L. Nemhauser. On the relation betweeen the traveling-salesman and the longest-path problems. *Operations Research*, 10(5):647–657, Sept-Oct 1962. 185

[101] Masayasu Hata and Shigeyuki Doi. Propagation tests for 23 GHz and 40 GHz. *IEEE Journal on Selected Areas in Communications*, 1(4):658–673, September 1983. 78

[102] Paul S. Heckert and Michael Garland. Survey of polygonal surface simplification techniques. In *Proc. 24th International Conference on Computer Graphics and Interactive Techniques*, May 1997. 240

[103] J.K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6):21–32, December 1994. 38

[104] Mordechai I. Henig. The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2):281–291, May 1985. 188

[105] Benjamin E. Henty. Throughput measurements and empirical prediction models for IEEE 802.11b wireless LAN (WLAN) installations. Master's thesis, Virginia Polytechnic Institute and State University, August 2001. 47

[106] Jeffrey Hightower, Anthony LaMarca, and Ian Smith. Practical lessons from Place Lab. *IEEE Pervasive Computing*, 5(3):32–39, July-September 2006. 66

[107] Jeffrey Hightower, Roy Want, and Gaetano Borriello. SpotON: An indoor 3D location sensing technology based on RF signal strength. UW CSE 00-02-02, Department of Computer Science and Engineering, University of Washington, Seattle, WA, February 2000. 64

[108] Harry Holma and Antti Toskala, editors. *WCDMA for UMTS*. John Wiley & Sons, 2nd edition, 2002. 54, 56

[109] Andy Hopper. The Royal Society Clifford Paterson lecture, 1999 - sentient computing. *Philosophical Transactions of the Royal Society, London A*, 358:2349–2358, August 2000. 30

[110] A. K. M. Mahtab Hossain, Hien Nguyen Van, Yunye Jin, and Wee-Seng Soh. Indoor localization using multiple wireless technologies. In *Proc. IEEE MASS*, pages 1–8, October 2007. 65, 77

[111] E. Huang, W. Hu, J. Crowcroft, and I. Wassell. Towards commercial mobile ad hoc network applications: A radio dispatch system. In *Proc. ACM MobiHoc*, pages 355–365, May 2005. 50

[112] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen K. Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. CarTel: a distributed mobile sensor computing system. In *Proc. ACM SenSys*, November 2006. 40, 70

[113] IEEE 802.20 Working Group. *IEEE Draft Standard P802.20 (D4.1m) for Local and Metropolitan Area Networks – Standard Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility – Physical and Media Access Control Layer Specification*. IEEE, 2008. 52

[114] IEEE Working Group 11. *IEEE Std 802.11-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Number ISO/IEC 8802-11:1999(E). IEEE, 1999. 56

[115] IEEE Working Group 11. *IEEE Std 802.11-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band*. IEEE, 1999. 57, 104

[116] IEEE Working Group 11. *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*. IEEE, 2003. 57

[117] Masugi Inoue, Khaled Mahmud, Homare Murakami, and Mikio Hasegawa. Novel out-of-band signalling for seamless interworking between heterogeneous networks. *IEEE Wireless Communications*, 11(2):56–63, April 2004. 63

[118] ISO. Road vehicles – controller area network (CAN) – part 1: Data link layer and physical signalling. Technical Report 11898-1:2003, International Standards Organisation, November 2003. 71

[119] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. Technical Report RFC 3775, IETF, June 2004. 59

[120] Alan Jones and Martin Brown. Choice routing: Technical overview. Technical report, Camvit Ltd., 2006. 160

[121] Kipp Jones, Ling Liu, and Farshid Alizadeh-Shabdiz. Improving wireless positioning with look-ahead map matching. In *Proc. MobiQuitous*, August 2007. 203

[122] L. R. Ford Jr. and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962. 182

[123] Silke Jung. HGV tolls in Germany: Innovative, environmentally friendly and fair. In *Proc. IEE Road Transport Symposium*, pages 5/1–5/7, December 2005. 51

[124] Marko Jurvansuu, Jarmo Prokkola, Mikko Hanski, and Pekka Perälä. HSDPA performance in live networks. In *Proc. IEEE ICC*, pages 467–471, June 2007. 59

[125] T. Kadonaga and K. Abe. Comparison of methods for detecting corner points from digital curves. In Rangachar Kasturi and Karl Tombre, editors, *Proc. 1st International Workshop on Graphics Recognition: Methods and Applications*, volume 1072 of *LNCS*, pages 23–34, August 1995. 132

[126] Kamol Kaemarungsi and Prashant Krishnamurthy. Properties of indoor received signal strength for WLAN location fingerprinting. In *Proc. MobiQuitous*, pages 14–23, 2004. 77

[127] Theodoros Kamakaris and Jeffrey V. Nickerson. Connectivity maps: Measurements and applications. In *Proc. 38th Hawaii International Conference on System Sciences*, pages 307–311, January 2005. 126, 129, 164

[128] M. Kamenetsky and M. Unbehaun. Coverage planning for outdoor wireless LAN systems. In *Broadband Communications. International Zurich Seminar on Access, Transmission, Networking*, pages 49–1–49–6, February 2002. 93

[129] Eiman Kanjo and Peter Lanshoff. Mobile phones to monitor pollution. *IEEE Distributed Systems Online*, 8(7), 2007. 41, 70, 156

[130] Hillol Kargupta, Ruchita Bhargava, Kun Liu, Michael Powers, Kakali Sarkar, Martin Klein, Mitesh Vass, and David Handy. VEDAS: a mobile and distributed data stream mining system for real-time vehicle monitoring. In *Proc. 4th SIAM International Conference on Data Mining*, pages 300–311, 2004. 41

[131] N. Karlsson. Floating car data deployment & traffic advisory services. In *Bridging the European ITS Business Cooperation with China*, December 2003. 40

[132] C. Peter Keller. Algorithms to solve the orienteering problem: A comparison. *European Journal of Operational Research*, 41(2):224–231, July 1989. 189

[133] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Proc. IEEE ICDM*, pages 289–296, 2001. 240

[134] Mark Klerer. Introduction to IEEE 802.20. Presentation to 802.20 WG, March 2003. 52

[135] Yin Fern Ko, Moh Lim Sim, and Maziar Nekovee. Wi-Fi based broadband wireless access for users on the road. *BT Technology Journal*, 24(2):123–129, April 2006. 51

[136] R. Koodli. Fast handovers for Mobile IPv6. Technical Report RFC 4068, IETF, March 2003. 60

[137] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless network research. Technical report, Dartmouth College Computer Science Department, July 2003. 124

[138] P. Krebs and U. Balmer. Fair and efficient: The distance related heavy vehicle fee (HVF) in Switzerland. Technical report, DETEC Switzerland, December 2004. 51

[139] D. G. Krige. A statistical approach to some mine valuations and allied problems at the Witwatersrand. Master's thesis, University of Witwatersrand, 1951. 126

[140] R. Kudwig and A. Gurtov. The Eifel response algorithm for TCP. Technical Report RFC 4015, IETF, February 2005. 61

[141] Markus G. Kuhn. An asymmetric security mechanism for navigation signals. In Jessica Fridrich, editor, *Proc. Information Hiding: 6th International Workshop*, volume 3200 of *LNCS*, May 2004. 49

[142] Houda Labiod, Hossam Afifi, and Costantino De Santis. *Wi-Fi^{TM} Bluetooth^{TM} ZigBeee^{TM} and WiMax^{TM}*. Springer, 2007. 53

[143] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device positioning using radio beacons in the wild. In *Proc. IEEE PerCom*, volume 3468 of *LNCS*, pages 116–133, 2005. 64, 128

[144] J. Lansford, R. Nevo, and B. Monello. Wi-Fi and Bluetooth: Enabling coexistence. *Compliance Engineering*, 18(4):30–45, May/June 2001. 44

[145] Henry Larkin. Wireless signal strength topology maps in mobile adhoc networks. In *Proc. Embedded and Ubiquitous Computing Conference*, volume 3207 of *LNCS*, pages 538–547, 2004. 127, 128

[146] Duan-Shin Lee and Yun-Hsiang Hsueh. Bandwidth-reservation scheme based on road information for next-generation cellular networks. *IEEE Transactions on Vehicular Technology*, 53(1):243–252, January 2004. 121

[147] Uichin Lee, Eugenio Magistretti, Mario Gerla, Paolo Bellavista, and Antonio Corradi. Dissemination and harvesting of urban data using vehicular sensor platforms. *Submitted to IEEE Transactions on Vehicular Technology*, 2008. 41

[148] Uichin Lee, Joon-Sang Park, Eyal Amir, and Mario Gerla. FleaNet: a virtual market place on vehicular networks (invited paper). In *Proc. 3rd Annual International Conference on Mobile and Ubiquitous Systems – Workshops*, pages 1–8, July 2006. 48

[149] Uichin Lee, Biao Zhou, Mario Gerla, Eugenio Magistretti, Paolo Bellavista, and Antonio Corradi. Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wireless Communications*, 13(5):52–57, October 2006. 41

[150] B. Li, J. Salter, A.G. Dempster, and C. Rizos. Indoor positioning techniques based on wireless LAN. In *Proc. 1st IEEE International Conference on Wireless Broadband & Ultra Wideband*, 2006. 65

[151] David G. Lowe. *Perceptual Organization and Visual Recognition*, chapter The Segmentation of Image Curves, pages 51–70. Kluwer Academic Publishers, 1985. 132

[152] Stephan Lück, Christian M. Mueller, Michael Scharf, and Robert Fetscher. Algorithms for hotspot coverage estimation based on field strength measurements. In *Proc. IEEE VTC*, pages 1086–1090, April 2007. 126, 152

[153] Stephan Lück, Michael Scharf, and Gil Jorge. An architecture for acquisition and provision of hotspot coverage information. In *Proc. 11th European Wireless Conference 2005*, pages 287–293. VDE Verlag, April 2005. 126

[154] Jun Luo and Jean-Pierre Hubaux. A survey of inter-vehicle communication. Technical Report IC/2004/24, School of Computer and Communication Sciences, EPFL, Switzerland, 2004. 51

[155] Ratul Mahajan, John Zahorjan, and Brian Zill. Understanding WiFi-based connectivity from moving vehicles. In *Proc. ACM IMC*, pages 321–326, October 2007. 77, 108, 123

[156] K. Maksuriwong, V. Varavithya, and N. Chaiyaratana. Wireless LAN access point placement using a multi-objective genetic algorithm. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1944–1949, October 2003. 93

[157] Glenford Mapp, David Cottingham, Fatema Shaikh, Edson Moreira, Renata Vanni, Wayne Butcher, Aisha El-safty, and Jon Crowcroft. An architectural framework and enabling technologies for heterogeneous networking. *Submitted to Journal of IEEE/ACM Transactions on Networking*, 2008. 24

[158] Glenford Mapp, David N. Cottingham, Fatema Shaikh, Pablo Vidales, Leo Patanapongpibul, Javier Balioisian, and Jon Crowcroft. An architectural framework for heterogeneous networking. In *Proc. International Conference on Wireless Information Networks and Systems (WINSYS)*, August 2006. 24, 236

[159] Gérard Maral. *VSAT Networks*. Wiley, 2nd edition, 2003. 51, 59

[160] Majed Marji, Reinhard Klette, and Pepe Siy. Corner detection and curve partitioning using arc-chord distance. In R. Klette and J. Žunić, editors, *Proc. IWCIA*, volume 3322 of *LNCS*, pages 512–521, 2004. 132

[161] J. McNair and F Zhu. Vertical handoffs in fourth-generation multinetwork environments. *IEEE Wireless Communications*, 11(5), June 2004. 63

[162] Hagit Messer. Rainfall monitoring using cellular networks. *IEEE Signal Processing Magazine*, 24(3):142–144, May 2007. 78

[163] Hagit Messer, Artem Zinevich, and Pinhas Alpert. Environmental monitoring by wireless communication networks. *Science*, 312:713, May 2006. 78

[164] Prashanth Mohan, Venkat Padmanabhan, and Ram Ramjee. TrafficSense: rich monitoring of road and traffic conditions using mobile smartphones. Technical Report MSR-TR-2008-59, Microsoft Research India, April 2008. 41

[165] Edson D. S. Moreira, David N. Cottingham, Jon Crowcroft, Pan Hui, Glenford E. Mapp, and Renata M. P. Vanni. Exploiting contextual handover information for versatile services in NGN environments. In *Proc. 2nd IEEE International Conference on Digital Information Management (ICDIM)*, volume 1, pages 506–512, October 2007. 24

[166] M. Moske, H. Fussler, H. Hartenstein, and W. Franz. Performance measurements of a vehicular ad hoc network. In *Proc. IEEE VTC*, volume 4, pages 2116–2120, May 2004. 48, 50

[167] P. Murphy, E. Welsh, and J. P. Frantz. Using Bluetooth for short-term ad hoc connections between moving vehicles: A feasibility study. In *Proc. IEEE VTC*, pages 414–418, May 2002. 53

[168] Chen Na, J. K. Chen, and T. S. Rappaport. Measured traffic statistics and throughput of IEEE 802.11b public WLAN hotspots with three different applications. *IEEE Transactions on Wireless Communications*, 5(11):3296–3305, November 2006. 47, 198

[169] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. TrafficView: Traffic data dissemination using car-to-car communication.

ACM Mobile Computing and Communications Review, 8(3):6–19, July 2004. 40

[170] Alok Nandan, Saurabh Tewari, Shirshanka Das, Mario Gerla, and Leonard Kleinrock. AdTorrent: delivering location cognizant advertisements to car networks. In *Proc. WONS*, pages 203–212, 2006. 48

[171] Vishnu Navda, Anand Prabhu Subramanian, Kannan Dhanasekaran, Andreas Timm-Giel, and Samir R. Das. MobiSteer: using steerable beam directional antenna for vehicular network access. In *Proc. ACM MobiSys*, June 2007. 62, 121, 194

[172] Maziar Nekovee. Sensor networks on the road: The promises and challenges of vehicular ad hoc networks and grids. In *Workshop on Ubiquitous Computing and e-Research*, May 2005. 47

[173] Jeffrey V. Nickerson. A concept of communication distance and its application to six situations in mobile environments. *IEEE Transactions on Mobile Computing*, 4(5):409–419, 2005. 195

[174] Washington Y. Ochieng, John W. Polak, Robert B. Noland, Lin Zhao, David Briggs, John Gulliver, Andrew Crookell, Ruthven Evans, Matt Walker, and Walter Randolph. The development and demonstration of a real time vehicle performance and emissions monitoring system. In *Submitted to 82nd TRB Annual Meeting*, July 2002. 70

[175] R. Olsen, D. Rogers, and D. Hodge. The aR^b relation in the calculation of rain attenuation. *IEEE Transactions on Antennas and Propagation*, 26(2):318–329, March 1978. 78

[176] IEEE 802.16 Working Group on Broadband Wireless Access. *IEEE Standard 802.16e-2005 for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*. IEEE, February 2006. 52

[177] J. Otero, P. Yalamanchili, and H.-W. Braun. High performance wireless networking and weather. Technical report, University of California, San Diego, 2001. 78

[178] Jörg Ott and Dirk Kutscher. The "drive-thru" architecture: WLAN-based internet access on the road. In *Proc. IEEE VTC*, May 2004. 108

[179] Jörg Ott and Dirk Kutscher. Drive-thru Internet: IEEE 802.11b for "automobile" users. In *Proc. IEEE INFOCOM*, March 2004. 108

[180] Jörg Ott and Dirk Kutscher. Why seamless? towards exploiting WLAN-based intermittent connectivity on the road. In *Proc. TERENA Networking Conference*, June 2004. 161

[181] Jörg Ott, Dirk Kutscher, and Mark Koch. Towards automated authentication for mobile users in WLAN hot-spots. In *Proc. IEEE VTC*, Fall 2005. 50

[182] M. Panjwani, A. Abbott, and T. Rappaport. Interactive computation of coverage regions for wireless communication in multifloored indoor environments. *IEEE Transactions on Communications*, 13(3), April 1996. 92

[183] K. Papagiannaki, M. Yarvis, and W. S. Conner. Experimental characterization of home wireless networks and design implications. In *Proc. IEEE INFOCOM*, pages 1–13, April 2006. 93

[184] Yang Byung Park and C. Patrick Koelling. A solution of vehicle routing problems in a multiple objective environment. *Engineering Costs and Production Economics*, 10(2):121–132, June 1986. 189

[185] Leo Patanapongpibul, Glenford Mapp, and Andy Hopper. An end-system approach to mobility management for 4G networks and its application to thin-client computing. *ACM SIGMOBILE Mobile Computing and Communication Review*, 10(3):13–33, July 2006. 61

[186] Bob Pearson. Complementary code keying made simple. Technical Report AN.9850.2, Intersil, November 2001. 57

[187] Kostas Pentikousis, Marko Palola, Marko Jurvansuu, and Pekka Perälä. Active goodput measurements from a public 3G/UMTS network. *IEEE Communications Letters*, 9(9):802–804, September 2005. 197

[188] Alex Pentland, Richard Fletcher, and Amir Hasson. DakNet: Rethinking connectivity in developing nations. *IEEE Computer*, 37(1):78–83, January 2004. 48

[189] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++*, chapter 14, pages 655–660. Cambridge University Press, 2 edition, 2002. 136

[190] Daji Qiao and Sunghyun Choi. Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation. In *Proc. IEEE ICC*, volume 7, pages 1995–2000, June 2001. 58, 96

[191] Maxim Raya and Jean-Pierre Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, 2007. 49

[192] Pablo Rodriguez, Rajiv Chakravorty, Julian Chesterfield, Ian Pratt, and Suman Banerjee. MAR: a commuter router infrastructure for the mobile Internet. In *Proc. ACM MobiSys*, pages 217–230, June 2004. 62

[193] Bogdan Roman, Frank Stajano, Ian Wassell, and David N. Cottingham. Multi-carrier burst contention (MCBC): Scalable medium access control for

wireless networks. In *Proc. IEEE Wireless Communications & Networking Conference*, pages 1667–1672, March 2008. 24

[194] Azriel Rosenfeld and Emily Johnston. Angle detection on digital curves. *IEEE Transactions on Computers*, C-22:875–878, September 1973. 131, 134

[195] Azriel Rosenfeld and Joan S. Weszka. An improved method of angle detection on digital curves. *IEEE Transactions on Computers*, C-24(9):940–941, September 1975. 131

[196] Paul L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):659–666, June 1997. 132

[197] Roy Rubenstein. Radios get smart. *IEEE Spectrum*, pages 46–50, February 2007. 237

[198] Abraham Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, July 1964. 136

[199] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. Technical Report RFC 3550, IETF, July 2003. 104

[200] Maria Grazia Scutella. An approximation algorithm for computing longest paths. *European Journal of Operational Research*, 148(3):584–590, August 2003. 185

[201] M. Sellars, G. Athanasiadou, B. Ziolko, S. Greaves, and A. Hopper. Simulation of broadband FWA networks in high-rise cities with linear antenna polarisation. In *Proc. IEEE PIMRC*, volume 1, pages 371–375, 2003. 43

[202] G. N. Senarath and D. Everitt. Comparison of alternative handoff strategies for micro-cellular mobile communication systems. In *Proc. IEEE VTC*, volume 3, pages 1465–1469, June 1994. 62

[203] Senza Fili Consulting. Fixed, nomadic, portable and mobile applications for 802.16-2004 and 802.16e WiMAX networks. Technical report, WiMAX Forum, November 2005. 52

[204] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. In *Proc. ACM MobiCom*, pages 334–345, 2006. 48

[205] Fatema Shaikh, Aboubaker Lasebae, and Glenford Mapp. Client-based SBM layer for predictive management of traffic flows in heterogeneous networks. In *Proc. IEEE ICTTA*, 2006. 152

[206] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. 23rd ACM national conference*, pages 517–524, 1968. 129

[207] Minho Shin, Arunesh Mishra, and William A. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *Proc. ACM MobiSys*, pages 70–83, June 2004. 63

[208] Anil Shukla, Aynur Alptekin, Julie Bradford, Eddie Burbidge, Derek Chandler, Mike Kennett, Paul Levine, and Stephan Weiss. Cognitive radio technology: A study for ofcom. Technical Report QINETIQ/06/00420, QinetiQ, February 2007. 237

[209] M. Siebert, M. Lott, M. Schinnenburg, and S. Gbbels. Hybrid information system. In *Proc. IEEE VTC*, page 5, May 2004. 121

[210] Jatinder Pal Singh, Nicholas Bambos, Bhaskar Srinivasan, and Detlef Clawin. Wireless LAN performance under varied stress conditions in vehicular traffic scenarios. In *Proc. IEEE VTC*, volume 2, Fall 2002. 107, 115

[211] Chris Snow and Serguei Primak. Performance evaluation of TCP/IP in Bluetooth based systems. In *Proc. IEEE VTC*, pages 429–433, May 2002. 53

[212] João Luís Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proc. ACM SIGCOMM*, pages 49–60, August 2003. 182

[213] W.-S. Soh and H. Kim. Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions. In *Proc. IEEE INFOCOM*, March 2004. 121

[214] H. Soliman, C. Catelluccia, K. El Malki, and L. Bellier. Hierarchical Mobile IPv6 mobility management (HMIPv6). Technical Report RFC 4140, IETF, June 2003. 59

[215] Mark Stemm and Randy H. Katz. Vertical handoffs in wireless overlay networks. *Mobile Networks and Applications*, 3(4):335–350, 1998. 58

[216] Steve Stroh. Wideband: multimedia unplugged. *IEEE Spectrum*, 40(9):23–27, September 2003. 59

[217] C. Sweet, V. Devarapalli, and D. Sidhu. IEEE 802.11 performance in an ad-hoc environment. Technical report, UMBC Department of Computer Science & Electrical Engineering, 1999. 96

[218] J. Sydor. A proposed high data rate 5.2/5.8 GHz point to multipoint MAN system. Technical report, IEEE, October 2000. 43

[219] James Tate. A novel research tool – presenting the highly instrumented car. *Traffic Engineering & Control*, 46(7):262–265, July 2005. 70

[220] Cho-Huak Teh and Roland T. Chin. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859–872, August 1989. 132, 133, 134

[221] Kai-Uwe Thiessenhusen. Implications on city traffic from floating car data. In *Proc. DAAD Summer School Traffic and Econophysics*, August 2003. 40

[222] Marc Torrent-Moreno, Daniel Jiang, and Hannes Hartenstein. Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks. In *Proc. ACM VANET*, pages 10–18, 2004. 109

[223] Mai Tran, George Zaggoulos, Andrew Nix, and Angela Doufexi. Mobile WiMAX: Performance analysis and comparison with experimental results. In *Proc. IEEE VTC*, pages 1–5, September 2008. 52

[224] Nishith D. Tripathi, Jeffrey H. Reed, and Hugh F. Van Landingham. Handoff in cellular systems. *IEEE Personal Communications*, 5(6):26–37, December 1998. 62

[225] T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809, September 1984. 189

[226] Ryuhei Uehara and Yushi Uno. Efficient algorithms for the longest path problem. In R. Fleischer and G. Trippen, editors, *Proc. International Symposium on Algorithms and Computation*, volume 3341 of *LNCS*, pages 871–883, December 2004. 185

[227] Ari Välsänen, Pekko Orava, and Henry Haverinen. Adaptive beacon interval in WLAN. European Patent EP1463242, September 2004. 96

[228] N. Van den Wijngaert and C. Blondia. A predictive low latency handover scheme for Mobile IP. In *Proc. International Conference on Mobile Computing and Ubiquitous Networking*, April 2005. 63

[229] U. Varshney. Vehicular mobile commerce. *IEEE Computer*, pages 116–118, December 2004. 39

[230] Héctor Velayos and Gunnar Karlsson. Techniques to reduce IEEE 802.11b MAC layer handover time. Technical Report TRITA-IMIT-LCN R 03:02, KTH, Royal Institute of Technology, Stockholm, Sweden, April 2003. 96, 104

[231] P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, and A. Hopper. Autonomic system for mobility support in 4G networks. *IEEE Journal on Selected Areas in Communications*, November 2005. 63

[232] P. Vidales, R. Chakravorty, and C. Policroniades. PROTON: A policy-based solution for future 4G devices. In *Proc. IEEE International Workshop on Policies for Distributed Systems and Networks*, June 2004. 63

[233] P. Vidales and F. Stajano. The sentient car: Context-aware automotive telematics. In *Proc. 1st European Workshop on Location Based Services*, September 2002. 70

[234] Pablo Vidales, Carlos J. Bernardos, Ignacio Soto, David Cottingham, Javier Baliosian, and Jon Crowcroft. MIPv6 experimental evaluation using overlay networks. *Computer Networks*, 51(10):2892–2915, July 2007. 24, 60, 63, 199

[235] R. Vijayan and J. M. Holtzman. Sensitivity of handoff algorithms to variations in the propagation environment. In *Proc. International Conference on Universal Personal Communications*, volume 1, pages 158–162, October 1993. 61

[236] Bernard H. Walke. *Mobile Radio Networks: Networking, Protocols and Traffic Performance*. Wiley, 2nd edition, January 2002. 53

[237] Arthur Warburton. Approximation of Pareto optima in multiple-objective shortest-path problems. *Operations Research*, 35(1):70–79, Jan.–Feb. 1987. 188

[238] Mark Weiser. The computer for the twenty-first century. *Scientific American*, pages 94–104, September 1991. 30

[239] Matthias Wellens, Burkhard Westphal, and Petri Mähönen. Performance evaluation of IEEE 802.11-based WLANs in vehicular scenarios. In *Proc. IEEE VTC*, pages 1167–1171, April 2007. 109

[240] Richard S. Whitt. Ex parte filing to the FCC. Technical Report FCC Proceeding 06-229, Google Inc., July 2007. 237

[241] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, Dec 2002. 88

[242] L. Wischhof, A. Ebner, and H. Rohling. Information dissemination in self-organizing intervehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):90–101, March 2005. 40

[243] Lars Wischhof, André Ebner, Hermann Rohling, Matthias Lott, and Rüdiger Halfmann. SOTIS – a self-organizing traffic information system. In *Proc. IEEE VTC*, volume 4, pages 2442–2446, April 2003. 40, 49

[244] G. Wölfle and F. Landstorfer. Dominant paths for the field strength prediction. In *Proc. IEEE VTC*, pages 552–556, May 1998. 92

[245] G. Wölfle, P. Wertz, and F. Landstorfer. Performance, accuracy and generalization capability of indoor propagation models in different types of buildings. In *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 1999. 93

[246] H. Wu, J. Lee, M. Hunter, R. M. Fujimoto, R. L. Guensler, and J. Ko. Simulated vehicle-to-vehicle message propagation efficiency on Atlanta's I-75 corridor. Technical report, Georgia Institute of Technology, November 2004. 50

[247] H. Wu, M. Palekar, R. Fujimoto, R. Guensler, M. Hunter, J. Lee, and J. Ko. An empirical study of short range communications for vehicles. In *Proc. ACM VANET*, pages 83–84, 2005. 108

[248] Wen-Yen Wu. A dynamic method for dominant point detection. *Graphical Models*, 64:304–315, 2003. 132, 133, 134

[249] Bo Xu, Ouri Wolfson, Channah Naiman, Naphtali D. Rishe, and R. Michael Tanner. A feasibility study on disseminating spatio-temporal information via vehicular ad-hoc networks. In *Proc. V2VCOM*, June 2007. 50

[250] G. Xylomenos and G. C. Polyzos. TCP and UDP performance over a wireless LAN. In *Proc. IEEE INFOCOM*, volume 2, pages 439–446, March 1999. 88

[251] G. Xylomenos, G. C. Polyzos, P. Mähonen, and M. Saaranen. TCP performance issues over wireless links. *IEEE Communications Magazine*, 39(4):52–58, 2001. 35

[252] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty. Performance evaluation of safety applications over DSRC vehicular ad hoc networks. In *Proc. ACM VANET*, pages 1–9, 2004. 109

[253] Faqir Zarrar Yousaf, Kai Daniel, and Christian Wietfeld. Performance evaluation of IEEE 802.16 WiMAX link with respect to higher layer protocols. In *Proc. ISWCS*, pages 180–184, October 2007. 59

[254] Moustafa Youssef and Ashok Agrawala. The Horus WLAN location determination system. In *Proc. ACM MobiSys*, June 2005. 65

[255] Jie Zhange, Henry C. B. Chan, and Victor C. M. Leung. A location-based vertical handoff decision algorithm for heterogeneous mobile networks. In *Proc. IEEE GLOBECOM*, pages 1–5, November 2006. 63

[256] Qing Zhao and Brian M. Sadler. A survey of dynamic spectrum access. *IEEE Signal Processing Magazine*, pages 79–89, May 2007. 237

[257] H. Zimmermann. *Innovations in Internetworking*, chapter OSI reference model – The ISO model of architecture for open systems interconnection, pages 2–9. Artech House, Inc., 1988. 236