Lemmon, Colin John (2010) ***Boundary mapping and its application to geographic routing*****. PhD Thesis, James Cook University.**

Boundary Mapping and Its Application to Geographic Routing

Colin John LEMMON BPsych(Hons) *JCU*

February 2010

A thesis submitted for the degree of
Doctor of Philosophy in the School of Business
James Cook University

Statement of Access

I, the undersigned, aut hor of this work, understand that James Cook University will make this thesis available for use within the University Library and, via the Australian Digital Theses network, for use elsewhere.

I understand that, as an unpublished work, a thesis has significant protection under the Copyright Act and;

I do not wish to place any further restriction on access to this work.

Signature   Date

Statement of Sources

DECLARATION

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Signature

Date

Acknowledgements

I would like to thank my current supervisors Dr. Ickjai Lee and Dr. Carrie Lui, and my past supervisors Dr. Phillip Musumeci, Dr. Douglas Maskell, Dr. Eoin Hyden and my co-supervisor Prof. Bill Lavery, for their support and assistance during this research.

I would also like to thank Dr. Chris Gaskett for his advice and proof reading, along with Assoc. Prof. Bruce Litow and Dr. Jason Holdsworth.

Abstract

Geographic routing is a self organizing, low overhead, distributed system for routing in ad-hoc wireless networks. Practical application of this approach is limited due to the lack of global information to deal with local minima at voids or the outer boundary. To address this problem, an improved geographic forwarding strategy Greedy-BoundedCompass was developed to reduce the instance of local minima. Greedy-BoundedCompass allows packets to move away from the destination without looping in situations where Greedy forwarding would fail. Greedy-BoundedCompass was applied to Greedy Perimeter State Routing (GPSR) to confirm its effectiveness as an alternate forwarding strategy.

The Boundary Mapping Protocol (BMP) was then developed to detect local minima, and probe boundaries; handling branches, edge crossovers, detecting probe home, and boundary confirmation. Using BMP, a multi-strategy Boundary State Routing protocol (BSR) was developed which incorporated Greedy-BoundedCompass forwarding. BSR manages boundary exit points, path selection for boundary traversal, swapping of boundaries, and loop prevention with multimode strategies. In response to performance issues, a low resolution grid occupancy mapping system was developed as a replacement for BMP to address excessive probe overhead and memory requirements.

Implementation, testing, and analysis of the improved geographic routing strategies

were performed using a purpose built network simulator. Metrics used included path completion rate, route efficiency, control overhead, and memory requirements.

Greedy-BoundedCompass reduced the number of local minima, improving the path completion rate of Greedy forwarding by 49.2% in sparse networks with a significant improvement in route efficiency of 8.9%. Greedy-BoundedCompass applied as a replacement for Greedy forwarding in GPSR also demonstrated a significant improvement in route efficiency. BSR then demonstrated a significant improvement in route efficiency over improved GPSR of 46.1% in sparse networks. The alternate low resolution grid occupancy mapping demonstrated a significant reduction in probe overhead and memory requirements compared to BMP.

Greedy-BoundedCompass forwarding has application in existing geographic routing protocols. BSR along with the low resolution grid occupancy mapping system is a promising approach to geographic routing with minimal local information maintained for routing around local minima. Future research will focus on refining the proposed grid occupancy mapping system and dealing with mobility.

List of Publications

Lemmon, C.; Lui, S. M.; Lee, I., Distributing Network Coverage Using Grid Mapping. *Submitted*.

Lemmon, C., Lui, S. M. & Lee, I., Review of Location-Aware Routing Protocols. *Advances in Information Sciences and Service Sciences*, vol 2, 2, pp. 132-143, June 2010.

Lemmon, C.; Lui, S. M.; Lee, I., Geographic Forwarding and Routing for Ad-Hoc Wireless Network: A Survey. *Fifth International Joint Conference on INC, IMS and IDC*. Seoul, Korea , pp. 188-195, August 2009.

Lemmon, C and Musumeci, P., Boundary Mapping and Boundary State Routing (BSR) in Ad-Hoc Networks. *IEEE Transactions on Mobile Computing*, vol 7, 1, pp. 127-139, January 2008.

Lemmon, C. and Musumeci, P., Cooperative Behaviour of Location Aware Nodes in Ad-hoc Networks. *Proceedings of IEEE DEST 2007*, pp. 512-515 Cairns, Australia, 2007.

Lemmon, C., Experimental Design, Modeling and Testing of Geographic Routing

Protocols, *Proceedings of SimTecT 2006, Melbourne,* Australia, 2006.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> *I have recorded my own impressions with great diffidence, claiming no more credit than may attach to an earnest desire to make myself useful, and to further geographical research.*
>
> *Charles Sturt*
> *Expedition into Central Australia, Chapter 1, 1948*

Ad-hoc wireless networks consist of an unstructured self organising peer to peer network architecture in which each and every node is a client, and any of these clients may be required to function as a router. These networks may form as a result of an impromptu need, such as emergency services, search and rescue, or in an unmanaged civilian environment such as a sporting event. In these situations, ad-hoc networks offer the advantage that they do not require the setup of servers or base stations, and because they are self configuring in nature they can be rapidly deployed without technical support.

The structure of an ad-hoc network may change randomly due to movement of a node, being turned on and off by the user, moving out of range of other nodes within the network, moving behind an obstruction such as a building, switching to sleep mode, battery failure, or through radio interference.

For routing, ad-hoc wireless networks employ modified versions of traditional routing strategies. These include distance vector, link state, and on-demand routing protocols.

Distance vector and link state protocols are termed global routing strategies as they maintain global information regarding the state of the network topology. Global routing strategies in ad-hoc networks are suitable for smaller networks with low mobility, but are limited in their scalability due to periodic and global dissemination of topology information.

On-demand routing protocols address the control overhead of distance vector and link state strategies through the use of a route discovery mechanism which initiates route queries only when a route is required. As a result, on-demand routing protocols consume minimal bandwidth in networks with low mobility. However, under conditions of high mobility and high traffic load, route requests are broadcast (flooded) through the network, which can consume excess bandwidth and limit network performance.

Cluster and hierarchical approaches address scalability by segmenting the network into zones or layers. These approaches incur excessive maintenance overhead for cluster header-election, cluster membership and hierarchical addressing, especially under conditions of random mobility.

Overall, none of the above ad-hoc routing approaches offers a complete solution in all environments in which these networks are deployed [1]. This is due to the wide range of factors such as network mobility, size, and traffic load, which affect the structure and dynamic nature of an ad-hoc wireless network.

The main limitations that constrain routing in ad-hoc wireless networks are limited bandwidth of wireless communication technologies and the resource limitations of many ad-hoc wireless devices (such as those in sensor networks). To address this problem, routing strategies must minimise the knowledge they maintain about other nodes and links within the network, and restrict the amount of control traffic required to exchange this information.

There is however an alternate approach to this problem which offers a low bandwidth and low latency solution to routing in ad-hoc wireless networks. This approach is called geographic (or Cartesian) routing. Geographic routing takes advantage of the location aware capabilities of personal computing devices and sensor network devices. This approach forwards information through physical space using the location of the destination, location of intermediate nodes, and their distance or direction in relation to the destination. This contrasts the traditional routing approaches which consider the network as a logical topology and maintaining node, edge, and path information for routing decisions.

Geographic routing has application in location aware sensor networks and with further development could be applied to ad-hoc wireless networks used in general applications, such as emergency services. The distributed nature of geographic routing offers the advantage of minimal control overhead and minimal latency without the need for routing table exchange or route discovery. As more devices become location aware there is a greater impetus for further research and development into the practical application of geographic routing to take advantage of these location aware capabilities.

Although geographic routing offers an alternative to traditional routing strategies, there are a number of practical limitations to the successful implementation of geographic routing in ad-hoc wireless networks which forms the framework for this thesis.

1.1 Clarification of Terminologies

For the purpose of this thesis we will use the term ***geographic forwarding*** to refer strategies that use location information to forward packets node by node towards the location of the destination and therefore provide a partial routing solution due to failure at a local minimum where no neighbouring nodes exist which are closer to the destination. We will use the term ***geographic routing*** for strategies that employ geographic forwarding, but also incorporate a backup routing strategy on geographic forwarding failure at a local minimum to provide 100% path completion (in connected networks).

1.2 Geographic Routing

Geographic routing protocols employ a basic geographic forwarding strategy to forward packets, node by node, towards the location of the destination, with intermediate next hop routing decisions based on selection of the neighbour which has the closest distance, compass setting, or some other measure of forward progress towards the destination.

Geographic forwarding has the advantage that it does not require dissemination of topology information, maintenance of routing tables, or route construction, prior to or during the forwarding process as in traditional routing approaches.

Geographic forwarding in this ***basic form*** offers a near stateless, low overhead, and low latency solution to routing in ad-hoc networks. It does not incur the overhead involved in building, maintaining, and distributing distance vector or link state routing tables, or incur the control overhead and latency of route discovery incurred by on-demand routing protocols.

Geographic forwarding has the advantage that it allows a packet to adapt to changes in the topology. If an intermediate node becomes unavailable the forwarding strategy simply selects the next best choice from its directly connected neighbours. Other benefits include the ability to weight individual next hop choices according to additional metrics. Routes can be altered node by node and packet by packet simply by considering additional Quality of Service (QoS) related parameters relating to the next hop neighbours, such as delay or available bandwidth [2].

Geographic forwarding in this form requires each node to know the current location of each directly connected neighbour (though strategies have been proposed to address this requirement). More importantly, geographic forwarding requires a location database which can provide the location of each required destination node beyond the immediate neighbours. For data aggregation this may not be a problem, but if all nodes are potential destinations then the overhead required to maintain this service could be significant.

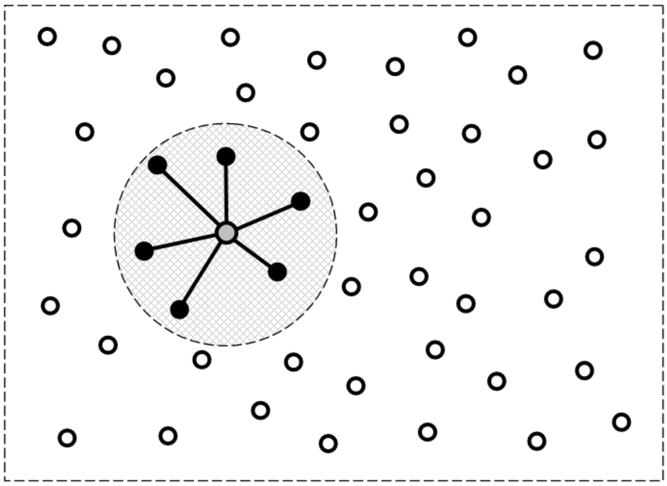

Figure 1: Information required by a geographic routing protocol.

The information a node must maintain for successful geographic routing is shown in Figure 1.

1.2.1 Neighbour Tables and Location Database

The following section explains in more detail the function of: 1) the neighbour table and 2) the location database.

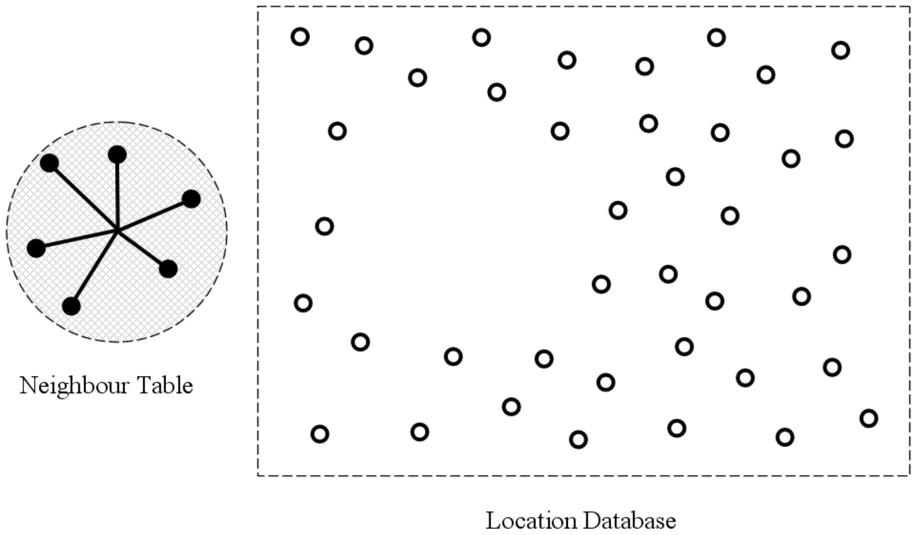

Figure 2: Components of a geographic routing protocol.

Figure 2 shows how the information in Figure 1 is managed by the neighbour table and the location database. The neighbour table contains the link state and location of all directly connected neighbours. This information is maintained through the periodic broadcast of beacons (called hello messages) between neighbouring nodes. Neighbour tables are not specific to geographic routing protocols and may be found in other non-geographic routing strategies, including link state protocols in wired networks such as Open Shortest Path First. For geographic routing the hello message will contain (at a minimum) the node ID (node identification number) and the node location.

The information in the neighbour table is used to select the best next hop candidate for a specific destination node based on the status of the link and the location of the neighbour in relation to the destination node location.

The location database in Figure 2 is specific to geographic routing and provides a list of destination node locations beyond the immediate neighbours. It is maintained as a service separate to the neighbour table [3] and may be run on a central server or implemented as a fully distributed system in an ad-hoc environment. In a distributed system the accuracy of the location information in the database will generally decrease as the distance of the remote destination node increases.

1.2.2 Route Failure at Local Minima

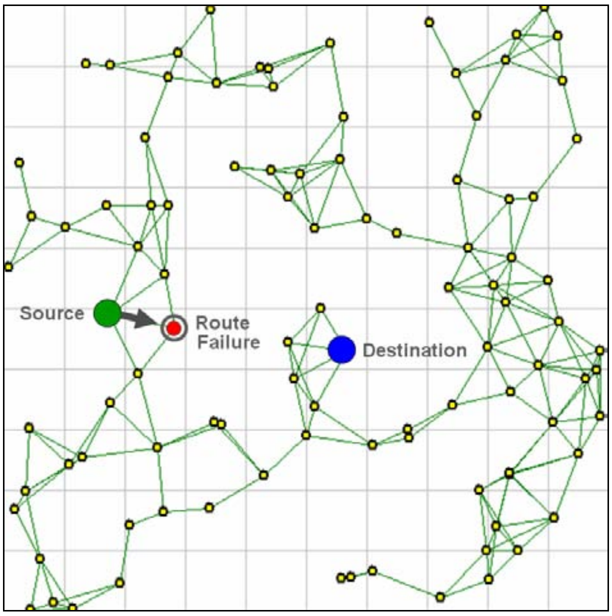

Figure 3: Geographic routing failure at a local minimum.

The inherent problem with geographic routing lies in the nature of ad-hoc networks.

The dependence of geographic forwarding on the physical network topography means that obstacles such as a building or lack of radio coverage may cause voids in the physical network topology or irregularities in the outer boundaries. This may result in local minima, where forwarding fails when a packet arrives at a node with no immediate neighbours that are closer to the destination (in terms of distance or some related measure) as shown in Figure 3.

When a packet encounters a local minimum, the geographic routing protocol must employ an alternate strategy to route the packet around the problem area. This will require either additional knowledge of the network topology, a search strategy, or the use of broadcasting (flooding). However, all of these strategies have the disadvantage that they add additional bandwidth and processing, and may increase latency.

From the network topology in Figure 3 it is evident that local minima can only occur on the boundary of a void in the network topology or on the outer boundary (otherwise there are links available for traversal in any direction). This fact can be used to identify specific nodes or localised areas where local minima may occur.

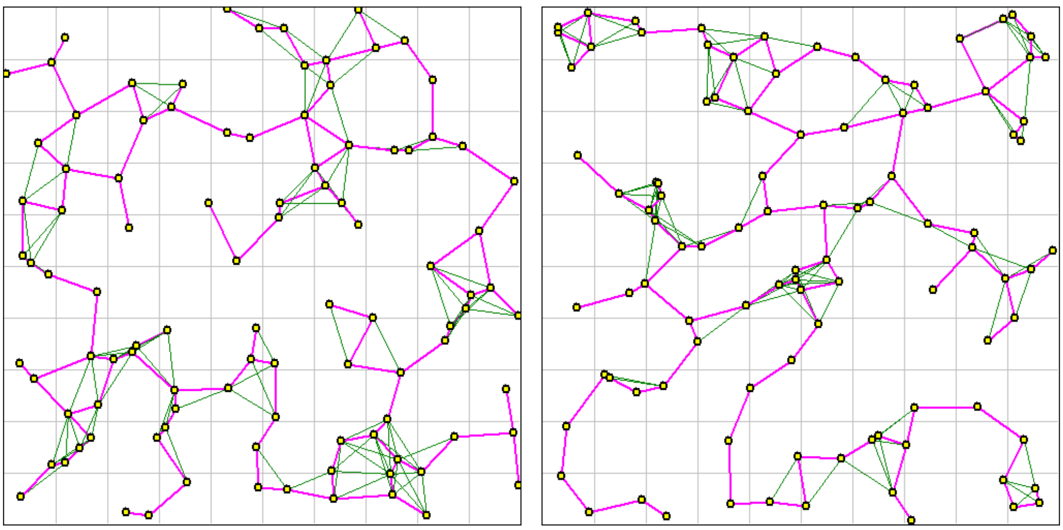

Figure 4: Diversity of boundary shape in sparse networks.

Figure 4 illustrates the characteristics of individual topologies which may vary considerably from one network to the next. In the first network the nodes are distributed around a large central void, whereas in the second network the nodes span out from the center in a spider like configuration. For a geographic routing protocol, each of these scenarios presents a different set of problems associated with local minima on inner voids and the outer boundary.

1.2.3 Geographic Routing Research

In networks with regular shaped boundaries and minimal voids, geographic forwarding can offer a low overhead routing option for moving data through a network. The challenge is to extend this advantage to more complex topographies by improving the efficiency of fallback strategies at local minima.

Research to date has had considerable focus on routing around local minima, but with limited success. Strategies proposed to address the problem of routing around local minima include restricted flooding [4], backtracking [5], planar graph conversion using face traversal [6], [7], [8], depth first search [2], [9] and hybrid approaches that incorporate conventional ad-hoc routing strategies [10], [11].

These strategies have not provided an adequate solution to the problem, and as result geographic routing is not a practical alternative for routing in ad-hoc networks even though the fundamental concept of geographic routing offers distinct benefits in terms of control overhead and latency.

1.3 Problem Statement

The problem to be addressed in this thesis is the inefficiency of current fallback strategies used by geographic routing protocols to route around local minima. This will require investigation into the balance between dependence on local knowledge only and proactively maintaining a minimal amount of global knowledge in specific areas where potential local minima arise.

As discussed earlier, it is a characteristic of local minima that they only occur on void boundaries or the outer boundary. In consideration of this an effective solution for the problem of routing around local minima should address the following points:

1. Minimise the instances of local minima
2. Isolate the nodes within the network where the remaining potential local

minima exist

3. Map the boundaries associated with these potential local minima.
    a. Distribute minimal link state information along alternate routes around local minima.
    b. Maintain sufficient but minimal information at a local minimum to make an optimal choice between alternate paths.
    c. Maintain sufficient but minimal information at a boundary node to make an optimal choice regarding swapping between boundaries or swapping to an alternate path.

The challenges in achieving these goals are minimisation of header control bits, minimisation of additional types of control traffic, and minimisation of the frequency of control traffic.

1.4 Aims

There are two main aims for this study.

1. To minimise local minima by improving the effectiveness of the basic geographic forwarding strategies. This will reduce the instances where a more complex (and less efficient) routing strategy is required for routing around local minima.
2. To investigate minimal approaches to boundary probing (in regards to probe initiation and probe overhead) and boundary mapping (in regards to memory requirements for data structures).

1.5 Research Questions

1. Can the path completion rate of the basic geographic forwarding algorithms be improved?
2. Will a forwarding strategy with improved rates of path completion improve the performance of an existing geographic routing protocol?
3. Can a geographic forwarding strategy be improved so that it can deal with potential local minima which occur at all nodes having a non reflex angle?
4. Can boundaries containing local minima be probed and mapped efficiently and effectively using limited (local) knowledge of the network topology?

5. Can boundary state information be used to improve the performance of a geographic routing protocol?

1.6 Limitations and Assumptions

This research is limited in regards to testing of protocols. Tests are only performed using a fixed network size of 1km by 1km with a fixed node count of 100 nodes (with a random distribution and node density varied by using set radio ranges). Future extensions to this research will require larger topologies and more varied distributions; however, these limitations are acceptable for the scope of this study.

The use of static networks is also a limitation of this study. This again will be addressed in future research as development needs to be extended to refine the algorithms used by the grid occupancy mapping, in addition to investigating mechanisms to segment the network to limit the range of global discovery and distribution.

Another limitation is that the custom network simulation software uses an idealised Media Access (MAC) layer and cannot adequately evaluate the degradation in performance due to latency and congestion. This has been evaluated in terms of probe and hello message overhead which again is within the scope of the study and will be addressed further when the functionality of the mapping is further developed.
There is an assumption by some existing geographic routing protocols and the Simple Boundary Mapping Protocol (BMPs), that the network can be adequately represented as a unit graph. Testing with the improved Boundary Mapping Protocol (BMP) was done using a quasi planar graph; however, this is still not realistic. The low resolution approach of proposed grid occupancy mapping offers the most promising solution to this problem and again is targeted for future research.

For all testing the existence of an ideal location database service is assumed.

1.7 Contributions

The main contributions of this study are:

1. The improved geographic forwarding strategy Greedy-BoundedCompass which has been shown to outperform Greedy forwarding in path completion with limited cost in terms of path efficiency.

2. BMP which will probe and distribute boundary state information, and given a destination location will provide the next hop for use by a geographic routing protocol. This protocol has limitations but some aspects will be useful in the continued development of the occupancy grid approach detailed below.

3. The Geographic Routing Protocol - Boundary State Routing (BSR). This protocol uses the improved geographic forwarding strategy Greedy-BoundedCompass along with boundary state routing decisions from BMP.

4. Grid Occupancy Mapping. To address limitations of BMP a low resolution, low overhead, mapping strategy is presented, which is more appropriate for future development to address issues of scalability and mobility.

More generally there is the investigation into the identification of local minima and related boundaries, the balance between local and global knowledge in relation to cost and performance, addressing looping when using multiple routing strategies that have no knowledge of the behavior and route history of the associated strategies, and the development of network simulation and graph/analytical software.

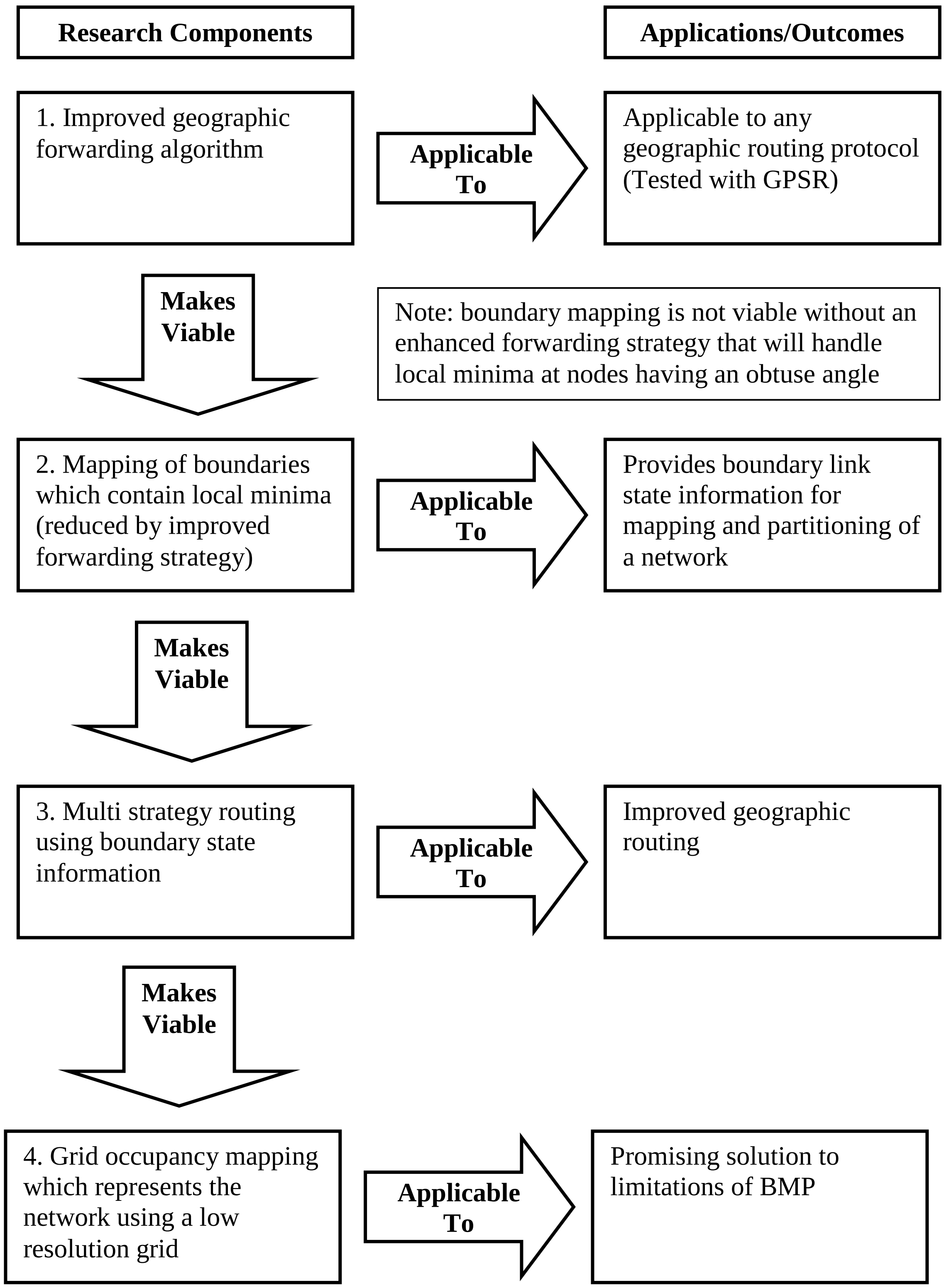

Figure 5: Steps proposed in the development and application of boundary mapping.

Figure 5 shows the dependencies between the components of the proposed solution to the development of a boundary mapping strategy and the application of the components to geographic routing.

1.8 Thesis Outline

Chapter 1: Introduction

This chapter provides a background to ad-hoc wireless networking and geographic routing with an explanation of the advantages and disadvantages of geographic forwarding including the problem of local minima. The components and services required for geographic routing protocol are explained for clarification. The problem statement is presented along with the research questions.

Chapter 2: Literature Review

This chapter provides an overview of non-geographic routing protocols and a discussion of the limitations of geographic routing with a focus on node location mechanisms and destination location services. Factors for classifying and discussing geographic forwarding strategies and geographic routing protocols are presented. Geographic forwarding strategies are discussed and compared, followed by a critical comparison of geographic routing protocols categorised as: Geographic aided routing protocols, flooding and backtracking, planar graph, boundary probing, waypoints (hybrid), zone based, power management, and stateless (beaconless) geographic routing. The balance between local or global knowledge is discussed as a framework for the thesis.

Chapter 3: Geographic Forwarding

This chapter proposes the BoundedCompass forwarding strategy and the multi-strategy BoundedCompass-Greedy and Greedy-BoundedCompass. Testing compares basic Greedy, MFR and Compass Forwarding with the bounded and multi-strategy approaches in relation to path completion and performance. This chapter also applies the multi-strategy Greedy-BoundedCompass forwarding to GPSR as a replacement for the Greedy forwarding component along with modifications to the GPSR algorithm. The protocols are tested and the performance improvement of the improved algorithm is discussed.

Chapter 4: Boundary Mapping Protocol (BMP)

This chapter outlines the initial algorithm for probing and mapping boundaries. This includes probe forwarding, discovery phase, branches, edge crossovers (looping), detecting probe home, and the boundary confirmation phase. An improved 2 hop neigh version is discussed which deals with non-uniform radio ranges. Testing evaluates probe overhead, additional hello message overhead, and memory overhead for boundary data structures.

Chapter 5: Boundary State Routing (BSR)

This chapter presents the geographic routing protocol BSR which uses Greedy-BoundedCompass forwarding along with boundary state information from BMP for routing around local minima. It discusses the calculation of the boundary exit point, path selection for boundary traversal, swapping of boundaries, loop prevention with multimode strategies, and protocol implementation (headers, data structures and pseudo code). BSR is tested and compared to both GPSR, and GPSR with Greedy-BoundedCompass. Results are discussed in relation to the compromise between maintaining local and global knowledge.

Chapter 6: Grid Occupancy Mapping

This chapter proposes the concept of a low resolution grid occupancy map to minimise probe overhead and memory requirements inherent in BMP. A trusted local map advertisement mechanism is outlined which extends the hello protocol. Optimal cell size is evaluated and a window based analysis is presented which deals with the complexity and variability of home detection. Testing evaluates probe overhead, additional hello message overhead, and memory overhead for boundary data structures.

Chapter 7: Experimental Design

This chapter discusses decisions relating to the experimental design and testing. It also outlines custom built simulation and graph/analytical software for testing.

Chapter 8: Conclusion

Relates the thesis body to the research questions and Hypothesis and proposes directions for future work.

Chapter 2

Literature Review

It has, therefore, been judged, that a succinct history of these discoveries would be acceptable to the public; and would form an appropriate introduction to a voyage, whose principal object was to complete what they had left unfinished.

Matthew Flinders
A Voyage To Terra Australis Volume I, Introduction, 1814

This chapter will review current and previous research relating to ad-hoc routing protocols. This will include an introduction to routing in ad-hoc wireless networks, an overview of conventional ad-hoc routing strategies, and a detailed review of geographic routing protocols.

2.1 Non-Geographic Ad-Hoc Routing

Traditional routing strategies used in wired networks have been adapted for use in ad-hoc networks. These conventional routing strategies view the network as a graph G(N, $E(t)$) consisting of a set of mobile nodes N (hosts) and edges E (undirected links) with their associated link cost (at one specific point in time t).

Conventional ad-hoc routing strategies include table driven, on-demand and hybrid approaches. Table driven routing strategies such as distance vector [12], [13] and link state [14], [15] routing protocols are suitable for smaller networks with low mobility, but do not scale well in larger dynamic environments due to the periodic and global dissemination of topology updates. On-demand routing protocols [16], [17], [18], [19], [20], [21] use a query response mechanism to discover and maintain routes for individual sessions. This addresses the control overhead of distance vector and link state strategies. However, flooding of route queries limits performance under conditions of high mobility and high traffic loads. The hybrid approaches [22], [23], [24], [25] use a cluster or hierarchical network structure to dynamically group nodes and then apply different routing strategies within and between groups. This addresses scalability in static networks or situations involving group mobility, but incurs excessive maintenance overhead for cluster head election, cluster membership and hierarchical addressing, under conditions of random mobility.

2.1.1 Table Driven Wireless Routing Protocols

Table driven routing protocols were adapted from traditional wired network routing protocols and include modified versions of both distance vector and link state routing protocols.

2.1.1.1 Distance Vector Wireless Routing Protocols

Distance Vector protocols in ad-hoc wireless networks operate in a similar way to those in fixed infrastructure wired networks such as Routing Information Protocol (RIP). These protocols which include Destination-Sequenced Distance Vector (DSDV) [12] and Wireless Routing protocol (WRP) [13] maintain a routing table within every node of the network consisting of a matrix of vectors to all known and available destinations. The routing table contains a list of destination addresses, the associated next hop address for the shortest path to the destination, and the distance (hop count) of the shortest path determined according to the Distributed Bellman-Ford (DBF) algorithm. This approach however suffers from the problem of routing loops (count to infinity) and slow convergence. Due to the dynamic network structure in mobile environments this can not be addressed using split horizon or poisoned reverse as in wired networks. DSDV and latter protocols addressed this problem by stamping each route update with a sequence number so that any node that receives the same update via an alternate path can recognise the update and drop the packet.

2.1.1.2 Link State Wireless Routing Protocols

Link State protocols propagate and maintain the full network topology to all nodes in the network. Link state protocols therefore provide multiple paths to a destination, converge more rapidly in static networks compared to distance vector protocols and do not create routing loops. However implementing link state protocols in a mobile wireless network is problematic due to the excessive bandwidth overhead incurred by the flooding of topology updates, especially in dynamic network environments which may change too rapidly for the routing tables to converge. This problem was addressed in GSR [14] by eliminating the flooding of topography changes and distributing information in a similar manner to distance vector protocols. FSR [15] proposed a further improvement to the overhead of topology updates by increasing the time between updates for destinations in proportion to the distance of the node from the destination. Using this strategy, accurate local link state information is maintained locally (3 hop radius) with the accuracy reducing as the distance from the node increases. When packets are routed through the network the link state information becomes more accurate as the packet approaches the destination.

Even with strategies to minimise control overhead of topology updates, both distance vector and link state routing protocols do not scale well to larger networks, and become less efficient if the network is in a high state of mobility or suffers increased node dropouts. This is due to requirements of periodic advertisements, the global dissemination of topology information, and the continuous flooding of table updates through the entire network when the network structure is dynamic. Excessive bandwidth requirements are compounded by the overhead involved in propagating and maintaining redundant information for unused routes. In dynamic environments these factors may result in interference, congestion, reduced bandwidth and packet loss. The resulting packet loss may delay table updates, increasing latency, and result in slower convergence and inaccurate or stale routing table entries. The more dynamic the network the more frequent will be the loss of links during a session and it is under these conditions where the network infrastructure is highly dynamic that accurate, up-to-date link topology information is the most critical.

Global routing protocols are however effective for smaller networks or in localised areas within larger networks where the overhead of table maintenance can be kept proportionate to the available network bandwidth. For example DSDV [12] is an early distance vector wireless routing protocol which is effectively used in clustered routing schemes including CGSR.

2.1.2 On-Demand Routing Protocols

On-demand routing protocols including LMR[16], DSR [17], ABR [18], SSR [19], TORA [20], and AODV [21] are connection oriented, using point to point routing with each route established as a separate executed process. This contrasts the connectionless approach used in distance vector and link state protocols. In the on-demand routing strategies, routes are only established when requested and then either cached or erased when the session is complete. Control overhead is therefore reduced by eliminating the periodic propagation of topology information, processing overhead of initialising and maintaining routing tables and storage of redundant or unused routes. It also minimises the client resource overhead for processing, memory and transmitter power, and allows nodes to operate in sleep mode as no beaconing is required when the network is idle.

On-demand routes are established using a source initiated query response process. Route discovery involves the broadcast (flooding) or multicast of small route discovery packets containing the destination id. The destination replies to the first packet it receives with a reply packet containing a list of the intermediate nodes traveled by the route discovery packet. The route information is either returned to the source or a link reversal algorithm is used where the reply packet sets the appropriate route entries in the intermediate nodes on its return to the source.

On-demand routing protocols scale better to larger networks than global routing protocols when traffic and mobility is low. However, increased traffic and mobility can cause increased network contention and congestion due to the excessive overhead involved in flooding. When a link failure occurs, on-demand approaches must initiate a route maintenance procedure using localised flooding of route maintenance packets to establish a link around the broken section. If unsuccessful an error message will be propagated back to the source which will initiate a new route discovery procedure. In dynamic environments route maintenance can therefore consume significant bandwidth for flooding of route requests for link maintenance.

On-demand protocols also suffer from latency in route construction and reconstruction (especially when mobility and traffic is high) making it less appropriate for short repetitive sessions that are dropped and re-established. Latency can however be addressed by caching of routes.

As nodes do not have access to the additional link metrics available to the table driven approaches there is no link quality information for individual links or composite paths from source to destination. The lack of distributed information creates a problem in partitioned networks as there is no information available to indicate that a destination or set of destination nodes are not available. As a result route requests will be broadcast to these destinations until the request procedure times out or are terminated by the user.

Control bandwidth and latency in route construction can be reduced through the use of route caching [17], [21]. This allows previously established routes to be cached when a node establishes a route or overhears a route. The source may use a cached route or an intermediate node may return a cached route during the route discovery procedure. The problem is that without periodic review it can not be determined how long till the route is stale as this will vary according to network mobility.

2.1.3 Cluster And Hybrid Routing Protocols

The cluster and hybrid routing protocols including ZRP [22], CGSR [23], CEDAR [24], and HSR [25] allow for greater scalability but are still problematic in conditions of high mobility. In the clustered approach nodes organise themselves into clusters (also called cells or zones) and elect a cluster head to manage the cluster (like a pseudo base station). The cluster head manages link state information within the cluster and establish routes on behalf of the cluster members (possibly using a different routing algorithm for inter-cluster routing). To QoS issues CEDAR [24] extracts a core of high bandwidth nodes for the cluster heads which will provide reliable high-bandwidth inter-cluster routing. These approaches may also incorporate a hierarchical addressing scheme as used in ZRP [22] and HSR [25]. This provides scalability and lower latency in route construction at the expense of the overhead and complexity of managing the cluster hierarchy.

These approaches suit networks with low mobility or high group mobility where group membership corresponds with cluster membership. However, in networks with high random mobility there may be excessive control and processing overhead involved in cluster head election, core extraction, cluster membership, gateway channel scheduling and cluster address maintenance.

2.1.4 Evaluation of Non-Geographic Routing Protocols

Overall the non-geographic routing protocols do not scale well to larger ad-hoc wireless network topologies as devices are generally resource poor and in large scale networks the routing table size increases quadratically with an increase in network size and will reach a practical limit for memory. In addition, every individual link change must be propagated through to every node in the network before the routing tables converge and consume considerable bandwidth. The on-demand protocols address the routing table overhead issue; however, the omnidirectional flooding of route requests is not scalable due to increasing bandwidth and contention as network size increases. In addition, the larger the network the greater the latency in the request response process for route establishment and route maintenance.

2.2 Geographic Routing Protocols

Geographic routing protocols (based on geographic forwarding) have the advantage that they do not maintain routing tables or require route discovery. This reduces processing, complexity, memory and bandwidth because forwarding decisions are based on the location of the destination and the location of immediate (directly connected) neighbours. Geographic routing protocols do not need to converge before routing is possible and scale better than standard non-geographic routing protocols in wireless network topologies. In dense networks geographic routing offers a low overhead scalable solution to routing using only local information (if used in conjunction with a scalable destination node location service). However, geographic routing is less suited to sparse networks (with irregular shaped voids and outer boundary) due to instances of local minima in the geographic forwarding process and the lack of global information to make appropriate routing decisions.

To deal with the issue of routing around local minima geographic routing protocols consist of a primary geographic forwarding strategy, in addition to a secondary recovery strategy which is used when the primary forwarding fails. Strategies proposed to address the problem of routing through local minima include restricted flooding [4], backtracking [5], face traversal of a planar sub graph conversion [6], [7], [8], depth first search [2], [9] and hybrid approaches that incorporate conventional ad-hoc routing strategies [10], [11].

Because geographic forwarding offers advantages over other approaches to routing research in this area focuses on three main issues to provide a practical application for use in ad-hoc wireless networks (or sensor networks). These include routing around local minima, reducing control overhead of periodic beaconing to maintain local information (link status and neighbour location), and power conservation and management. Although there are a considerable number of proposed geographic routing strategies, currently none have been adopted for general use in ad-hoc wireless networks.

The following sections will provide details on the protocols and strategies proposed to handle the problems associated with geographic routing, along with a review of geographic routing protocols that address routing around local minima, minimisation of control overhead, and power management.

2.2.1 Limitations of Geographic Routing

Two fundamental issues that adversely effect the practical application of geographic routing are determining individual node location, and location service for dissemination of destination node locations.

2.2.2 Node Location Mechanisms

For all geographic routing protocols there is a requirement for nodes to know their location. Small embedded GPS devices are now available for this purpose although there are limitations. GPS accuracy is limited and is a function of occupation time and is affected by environmental conditions. Signal strength is dependent on the space and orientation for an aerial (which may vary in type and gain) and reliability is further affected by atmospheric conditions, environmental factors like foliage (especially if wet), and occupation time. Although this technology is improving continuously, the effect of extreme atmospheric conditions may always be a problem when there is insufficient signal available for processing. One option to address this issue is to install fixed or movable ground based GPS base stations for increased accuracy and reliability.

Other location mechanisms may also be employed. In proximity based systems a node can establish its location by analyzing its proximity to other nodes based on relative signal strength [26]. This system was shown to be effective in an open environment where all nodes have the same characteristics. However, this approach will not perform well in areas where propagation patterns are affected by the environment or devices are not of the same type (power, aerial characteristics etc.). An alternative to this approach is a virtual coordinate system that does not reflect the actual physical location of the nodes [27].

2.2.3 Destination Location Service

As discussed previously, the location database does not contain edge information like the topology database in link state routing protocols, and therefore requires considerably less overhead to disseminate and maintain. In addition, remote destination location can be less current, less frequently updated, and therefore less accurate than standard routing table information. Because geographic routing protocols reevaluated the route at each node, more accurate information will become available as the packet approaches the destination, and so appropriate routing choices will be made as the packet progresses along the route.

The problem with geographic routing is that destination node locations must be available at a minimum to a node that is instigating a route or a gateway node within a local cell that is initiating a route on behalf of the source node. Querying destination nodes directly is not scalable as these types of mechanisms would need to flood destination location queries as in [28] and [29]. The maintenance of location information requires either a centralised server, location proxies, or a distributed location database. Using a centralised server as a location database can not be assumed in an ad-hoc network environment where nodes operate independently and there is no guarantee that any one node has the resources and bandwidth capabilities to perform all memory, processing, and distribution for this service.

A fully distributed solution where all nodes hold location information for the entire network would be inefficient. A more appropriate and scalable solution is to use location proxies [30]. Alternatively, the Grid Location Service (GLS) [31] is a grid based hierarchical service that scales well, and reduces bandwidth by using local query within a cell and queries from the server of that cell to servers in higher order cells (levels) for remote queries. Because of its distributed nature of this strategy the system degrades well with node dropouts. Still there are issues with location update rate, bandwidth overhead, and accuracy and the effect of node velocity on more distant (older) routes though it was shown to perform well with node velocities between 0 and 10m/s with network sizes of 600 nodes.

[32] proposes correcting location errors originating from neighbour and destination mobility (called LLNK and LOOP errors) and propose a Neighbour Location Prediction (NLP) scheme and Destination Location Prediction (DLP) scheme to estimate locations more accurately in dynamic environments.

2.2.3.1 Location Service Elimination

Last Encounter Routing (LER) is an alternate approach used by Exponential Age SEarch (EASE) [33]. This strategy uses node mobility to disseminate node location information. Nodes cache information (time and location) of the last encounter with each of the other nodes in the network with which they have been in direct communication. If a node wants to send a packet it checks its cache for the destination nodes last time of encounter. EASE then uses LER to search surrounding nodes in an increasing radius until a node is found whose last encounter with the destination is less than or equal to half the time of the current nodes last encounter with the destination. The node that responded to the query is then used as a waypoint for routing (although no specific routing strategy is specified). This method will provide a rough estimate of location for distant nodes after which point in the route the destination location will get progressively more accurate as the node approaches the destination. An improved version Greedy EASE (GREASE) [33] is also proposed. In this version, if a node is encountered with a more recent estimate of the destination location than the waypoint then that location becomes the new waypoint.

2.2.3.2 Test Methodologies Relating to Location Service

Testing requires a decision whether to implement a location distribution strategy or use idealised location information. For comparison to non-geographic routing strategies or other geographic routing strategies that include a location mechanism, the inclusion of a location service would be mandatory. However, when comparing geographic routing strategies that are independent of the location service it is more controlled to compare routing strategies independent of the location service as overall performance could vary dependent on implementation choices. For this reason most testing uses an idealised location database and location information may be degraded in accuracy for testing.

2.3 Factors Effecting Geographic Routing Protocols

The following are limitations which vary between protocols.

- Flooding
- Reactive Searching
- Heterogeneity
- Memorisation
- Latency
- Traffic aggregation
- Scalability
- Power management

These are discussed in more detail below.

2.3.1 Flooding

Flooding invokes multipath routes and is therefore not scalable. Flooding consumes excess bandwidth and transmit power as nodes forward multiple copies of the same packet, which in resource poor low bandwidth environments is unacceptable. Nodes also waste additional processing power when they receive multiple copies of the same packet via different paths and disassemble the packets to determine whether they are duplicates and can be dropped. In addition to wasting valuable resources, flooding can increase latency when nodes are blanketed with signals that prohibit them from transmitting legitimate data packets until the unwanted duplicate transmissions subside.

The advantage of flooding is that it is suitable for heterogeneous environments and can guarantee delivery if the network is connected. However, a viable, scalable solution to routing in resource poor environments must avoid consuming unnecessary bandwidth and power through the use of flooding.

2.3.2 Reactive Searching

Some geographic routing protocols use a reactive mechanism to deal with failure at a local minimum and initiate a depth first search strategy (as apposed to a breadth first search as in flooding) to find an alternate path around the problem area. The disadvantage of this approach is the latency involved in establishing the route. In addition the maintenance of the route may require memorisation for individual data streams (which may or may not be a problem) and route maintenance if the discovered route becomes invalid. A more appropriate solution would be to proactively discover alternate routes around potential local minima in advance.

2.3.3 Heterogeneity

Many of the proposed solutions to geographic routing are based upon the concept of a unit graph which describes the idealised uniform circular radio range of radius R for all network devices. However, this is unrealistic in a practical wireless environment as all radio ranges and patterns of coverage would have to be equal and uniform. In a real world environment transmission range may vary between devices and may vary over time for a single device. There may also be holes in the pattern of coverage. Even if a transmitting device has a uniform coverage, receiving devices may be effected by interference from another source that is out of range of the transmitting device and hence the area of reception is distorted in respect to the transmit radius. An alternate partial planar graph and practical limits of coverage relating to this issue are discussed later.

2.3.4 Memorisation

Memory may be consumed for traffic redirection, destination location caching, waypoints, and cost metrics such as reliability and end point power levels.

Memorisation is only an issue if it becomes un-scalable as in the case of table driven routing protocols. However, if the resources are available (hardware and embedded system devices are improving in capability) memory availability may not represent a problem.

2.3.5 Latency

Latency is inherent in any strategy that must perform an on-demand search for a route or information related to the route. Latency is a problem in any network environment. Some strategies discussed below have inherent latency for every next hop choice and so accumulated latency may be a significant problem.

2.3.6 Traffic Aggregation

To address the limited bandwidth of wireless networks data traffic should be spread out where possible and not concentrated or aggregated in a subset of nodes in a backbone, waypoints, boundaries or through a subgraph. However this statement requires qualification.

Although unnecessary concentration of traffic needs to be considered, proactively diverting or splitting traffic without global knowledge of the network (and known, guaranteed alternate routes) could have negative consequences on path completion. Using congestion as a metric could be useful; however, splitting traffic may be detrimental unless the nodes of the alternate paths are separated enough so that their radio transmissions do not to interfere, otherwise the effects of interference and retransmission may be worse than using a single aggregated route.

2.3.7 Scalability

Scalability is limited by any factor that would increase resource consumption or degrade routing protocol performance as the size of the network increases. This may include flooding (bandwidth), searching (latency), memorisation, and reliance on global rather than local information.

2.3.8 Power Management

Power management is a desirable characteristic in ad-hoc wireless networks due to possible power supply limitations (battery life) although this is not the case in all applications (like vehicular sensor networks). Power consumption may be addressed by reducing transmit power to a level adequate for the current link traffic, considering the remaining energy levels of next hop neighbour candidates in forwarding decisions, and finally using a sleep management system for inactive nodes. These issues are discussed in detail later.

2.4 Geographic Forwarding Strategies

The fundamental routing strategy employed by geographic routing is geographic forwarding. In dense networks without voids and with regular shaped outer boundaries geographic forwarding can achieve guaranteed delivery. However any irregularity that creates the situation where there is no node closer to the destination (local minima) will cause the forwarding strategy to fail. Most reviews of geographic routing protocols incorporate geographic forwarding strategies in with routing strategies; however, this thesis considers that any ad-hoc network may have irregularities causing geographic forwarding to fail, so geographic forwarding is considered to be a component a practical geographic routing protocol.

The majority of geographic routing protocols uses Greedy forwarding, which forwards packets to the neighbour that is closest to the destination. However two alternate strategies have been proposed which are Most Forward within Fixed Radius R (MFR) [5] and Compass Routing (DIR) [34].

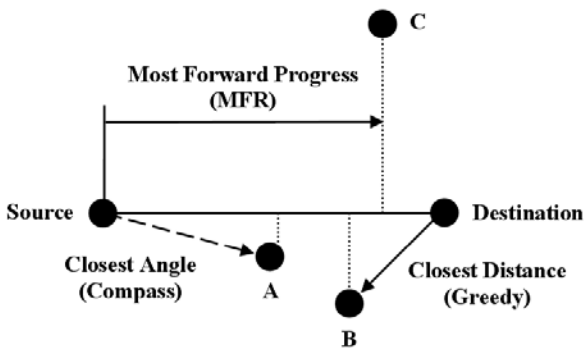

Figure 6: Greedy Forwarding, Compass Forwarding and Most Forward Progress.

2.4.1 Greedy Forwarding

Greedy forwarding [35] selects the next hop as the node closest to the destination. Because Greedy makes the largest possible movement towards the destination it will in most cases follow the shortest path. In sparse networks traversing to the node closest to the destination may involve some sideways deviation which could set the packet on a deviant path and there is no intelligent behavior or global knowledge available to avoid this situation.

At local minima Greedy forwarding is generally restricted from any backwards movement (away from the destination) to eliminate the possibility of looping. Flooding within a limited radius of *N* nodes is proposed as a recovery procedure at a local minimum. However, flooding even within a limited radius increases the bandwidth overhead.

Another characteristic of Greedy forwarding is that it allows a packet to move to a node that is beyond the destination if that node is closer to the destination than the previous node.

2.4.2 MFR

Most Forward within Fixed Radius R (MFR) [5] forwards packets to the neighbour within a set radius of the current node (not the route source) that makes the most forward progress (or the least backward progress) along the line drawn from the current node to the destination. Progress is calculated as the cosine of the distance from the current node to the neighbour projected back onto the line from the current node to the destination. Although typically reviewed as MFR, the authors suggest that it be implemented as Most Forward within N (MFN), which selects the next hop from the closest of *N* nodes, where the optimal value of *N* was found to be 7. It is important to note that this research focused exclusively on optimal transmission range, and the forwarding of a packet at a single node, with no consideration of the traversal of a packet along the entire path. MFR (or MFN) is not suitable for use as a geographic forwarding strategy as it is susceptible to looping. Because progress is measured from the current node to the destination, limiting progress to the forward direction only would not eliminate looping. Another disadvantage of MFR is that while a packet can progress towards the destination according to the measure of forward progress, it can continue to move away from the destination even though there are nodes that are physically closer or on a more direct trajectory.

The authors discuss options to extend the basic forwarding strategy using backtracking when packets reach a local minimum and cannot move forward. However backtracking is shown in later studies to introduce routing loops [4].

2.4.3 Compass Routing (DIR)

Compass Routing (DIR) [34] selects the neighbour on the closest angle to the destination. This results in a packet following the most direct trajectory from the source to the destination. Because Compass forwarding is not limited to traversal in the forward direction, it has the advantage that it can, in limited circumstances (discussed later), successfully progress around a boundary where the path moves away from the destination. This can result in a higher rate of path completion, but has the disadvantage that it makes Compass forwarding susceptible to routing loops.

The authors further propose that delivery can be guaranteed using face traversal of disjoint regions, where a packet is forwarded around one side of each face until the packet reaches the further edge of the face that intersects the line from the source to the destination. From this point the packet traverses the next face in a similar manner until the destination is reached. This is discussed in more detail later in this chapter.
The Random Compass algorithm [36] has been proposed for use as a routing strategy in convex planar sub graphs. This strategy selects the next hop randomly between the two nodes on the closest angle to the destination which are on either side (clockwise and anticlockwise) of the line from the source to destination. This algorithm has been shown to work for all convex subdivisions. A variation, Greedy-Compass [37] was also proposed which selects one of the two nodes which is at the minimum distance from the destination.

2.4.3.1 Compass Forwarding Around Convex Boundaries

The ability of Compass forwarding to progress around a boundary where the packet may move away from the destination is illustrated in Figure 7(a). With Greedy forwarding, the packet will traverse from A to B where it would be dropped as no node exists that is closer to the destination. Using Compass forwarding the packet will traverse from A to the node on the closest compass setting to the destination B. From B it will traverse around the boundary C, D, E, F and then to the destination as shown.

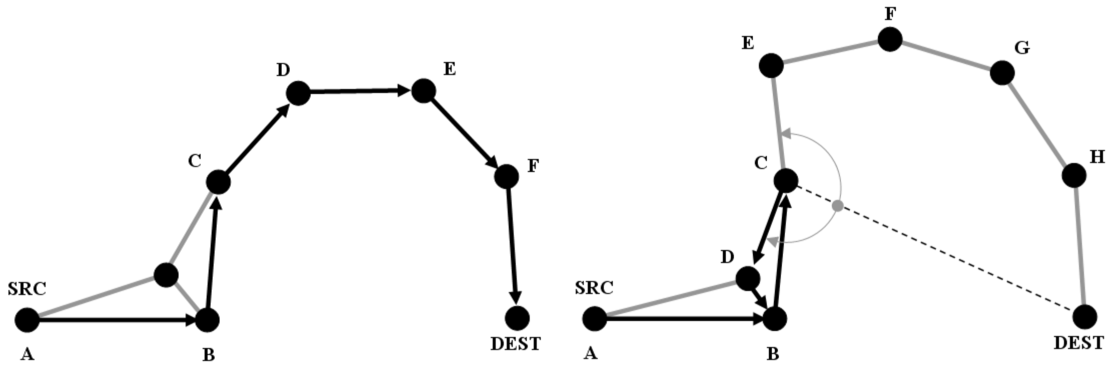

Figure 7: Compass forwarding (a) without looping and (b) with looping.

Figure 7(b) illustrates how Compass forwarding can result in a routing loop. In this example the packet will traverse from the source at A to B and then C as per the previous example. At C, D (not E) is on the closer compass heading to the destination and so the packet will traverse to D then to B, and will continue to loop around the path B, C, D until the packet is dropped.

2.4.4 Other Forwarding Strategies

Other strategies have been proposed that aid in the management of congestion when forwarding packets. The Random Progress Method [38] proposed forwarding packets to a random neighbour from among those that are closer to the destination. The random selection of the next hop offers the advantage of distributing the traffic load, however in comparison to the previous geographic forwarding strategies this approach does not use any measure of progress to differentiate any single candidate next hop as better than another.

2.5 Review of Geographic Routing Protocols

Geographic routing research in general focuses on improvements in route completion rates at local minima, in addition to minimising the power requirements and control overhead so that the advantage of geographic forwarding can be realised in a practical routing protocol. There are approaches that guarantee delivery and those that address problems using a probabilistic approach. For those that guarantee delivery a number of approaches are proposed for recovery at local minima including depth first searches and breadth first search strategies. The most popular approach is that of face traversal of a planar subgraph.

The following section provides a comparison of current geographic routing protocols for ad-hoc networks. Protocols are categorised according to the strategies they use to deal with local minima and the enhancements to improving their efficiency.

- Geographic aided routing protocols
- Searching: flooding and backtracking
- Planar graph
- Waypoint
- Zone based
- Power aware
- Stateless (Beaconless)

2.5.1 Background

Imielinsky and Navas [39] used the Greedy approach introduced by Finn [35], applying the concept of geographic routing to the Internet and proposed RFC2009 [40]. The proposal relates to geographic addressing and routing in a large scale cellular infrastructure. They identify the need for accessing location dependent data on the Internet and propose the integration of a geographic addressing scheme and related protocols into the Internet Protocol (IP). They further proposed a geographic messaging scheme where packets could be unicast or multicast into a geographic area defined by a circle or polygon. This research was instrumental in regenerating interest into geographic routing after a ten year period following Finn [35].

2.5.2 Geographic Aided Routing Protocols

Geographic aided routing protocols use the location of the current node and the destination to enhance the functionality of conventional ad-hoc routing strategies.

Location Aided Routing (LAR) [28] is an on-demand routing protocol that uses the last known position of the destination node and its velocity to limit the flooding of route requests towards the destination. Flooding is limited to an area between the source and a circle, calculated around the destination, with its center at the last known position and a radius which is determined by the node's velocity. This improves the efficiency of the underlying on-demand protocol but still suffers the problem of scalability and latency associated with on-demand strategies.

Distance Routing Effect Algorithm for Mobility (DREAM) [29] is based on the flooding of data without the prior establishment of a route. Messages are flooded into an area limited in a similar manner to that used in LAR. However, the use of directional flooding of data packets, as opposed to flooding of route requests in LAR, incurs additional bandwidth overhead. LAR has a small additional overhead of maintaining and distributing destination node velocity. This allows more accurate prediction of node location than DREAM. The main disadvantage of LAR over DREAM is that it is an on-demand protocol and therefore has the disadvantage of the latency involved in the query response (search) process for all routes and the overhead of route maintenance when the route is broken. This does provide an improvement on non-geographic ad-hoc on-demand routing strategies but at the cost of the additional overhead of maintaining destination node locations and velocities.

Both LAR and DREAM employ flooding which is not a scalable mechanism for geographic routing. These protocols, like DIR are based on direction. To target guaranteed delivery they must allow packets to move backwards at a local minimum. However backward progress without global knowledge or memorisation of path nodes and traffic may result in looping and will therefore not provide guaranteed delivery.

The following sections will discuss full geographic routing protocols which use geographic forwarding and a recovery strategy for failure at local minima. The protocols are categorised according to the recovery strategy employed.

2.5.3 Flooding and Backtracking (Searching) at Local Minima

Failure at a local minimum can be addressed using a search strategy to find an appropriate route. Flooding is a breadth first search and backtracking is a depth first search. Both these strategies were investigated in GEographic DIstance Routing (GEDIR) [4] which uses the Greedy forwarding strategy proposed by Finn [35] along with alternate recovery strategies.

To allow a packet to move through local minima, GEDIR does not include the current node in the distance calculation and permits a packet to backtrack in the reverse direction (away from the destination) if no forward node is available. To prevent looping the packet is not permitted to be passed from the neighbour back to the previous node. This addresses single hop looping but a packet may loop back via an alternate path making it unsuitable for practical application.

Two variations on GEDIR [4] were proposed to address the backtracking problem. These include flooding at local minima (*f*-GEDIR) and maintaining 2-hop neighbour information to predict and avoid local minima (*2-hop* GEDIR). *f*-GEDIR was found to be effective at the expense of increased control overhead while *2-hop* GEDIR was an improvement but still allows 2+ hop loops. A multi-path version *c*-GEDIR is also proposed to add reliability. *f*-GEDIR also seeks to reduce the incidence of local minima. Nodes that are concave flag this to their neighbours who drop them from their neighbour candidate list for that destination. This was shown to improve the performance in the tests performed; however, concave nodes do not flag local minima on indentations on the outer boundary and do not always flag local minima on the inner boundary. The flooding option is not an optimal solution and multipath may add reliability but may also cause interference and contention which can reduce bandwidth.

The above strategies seek to address the problem of local minima with local knowledge only - without any extra processing, memorisation, latency, control overhead or global knowledge of the network topology. This is in keeping with the simplicity of the geographic forwarding, however uninformed decisions at a local minimum require considerable bandwidth for flooding or ad-hoc undirected searches that do not seek optimal paths and are prone to looping.

In a more structured approach the Geographic Routing Algorithm (GRA) [9] proposes a flooding algorithm (breadth first search) but also proposes an alternative a depth first search. When a local minimum is reached a depth first search route discovery process is initiated to find a path all the way to the destination (not just to a point beyond the local minimum). The next hop is selected as the neighbour that has the lowest combined distance from source to the neighbour plus the neighbour to the destination. Looping is avoided by inserting node path information into the packet so that nodes are not revisited if alternate neighbours exist (otherwise the packet can backtrack) and the local minimum node ID is removed from the packet path list. Nodes cache routing information and progressively build up routing tables from the discovery procedures to improve efficiency and reduce unnecessary searches. These tables are then used in place of geographic forwarding when cached route information is available. This strategy is effective for static environments but suffers the latency inherent in on-demand searches, and has the additional problem of defining and managing stale route information.

All of the search strategies are heterogeneous and the full searches provide guaranteed delivery and optimal routes around local minima. However flooding (breadth first search) is not a scalable solution for ad-hoc wireless environments. Reactive searching adds unwanted latency but caching of routes may alleviate this, although managing the aging of information may be problematic. Memorisation of routes around local minima does not involve the magnitude of information as conventional table driven approaches and so may not be a problem.

2.5.4 Planar Graph

A commonly used strategy for routing around local minima is the face traversal of a planar subgraph [4], [7], [34], [41]. The planar subgraph (a graph that has no edge crossovers - that is, all edges intersect at the end points only) is extracted from the full network topology using a distributed algorithm requiring local knowledge only.

To reroute the packet at a local minimum the packet is forwarded along the first edge (in the planar graph) according to the left hand (anticlockwise rotation) or right hand rule (clockwise rotation) from the line drawn between the current node and the destination as illustrated in Figure 8. At each subsequent node the packet will be forwarded along the edges of the face with a set strategy for face change (discussed later). In the example in Figure 8 the packet is forwarded along the first edge on an anticlockwise rotation from the incoming edge unless the new edge crosses the line drawn between the current node and the destination at a point closer to the destination than the entry point to that face, in which case the next face is entered.

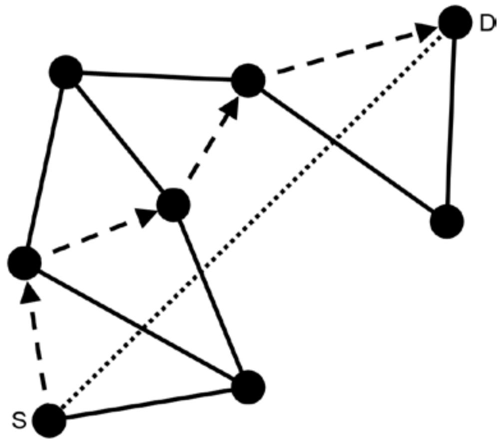

Figure 8: A face traversal algorithm for a planar graph.

For local construction of the planar subgraph two algorithms have been used – the Gabriel Graph (GG), Relative Neighbourhood Graph (RNG) and Delaunay triangulation as shown in Figure 9. For the GG, at node *u* the edge to neighbour *v* is retained if no other nodes exists within the circle drawn between *u* and *v*. For the RNG, at node *u* the edge to neighbour *v* is retained if the distance to *v* is less than or equal to the distance from both *u* and *v* to every other node, otherwise the edge is dropped. This is illustrated in Figure 9: where the edge (*u*,*v*) will be retained if no nodes exists within the shaded area which represents the area within d(*u*,*v*) from both *u* and *v*. For the Delaunay triangulation the edges are retained of the circle intercepting the triangle edge end points contains no other nodes.

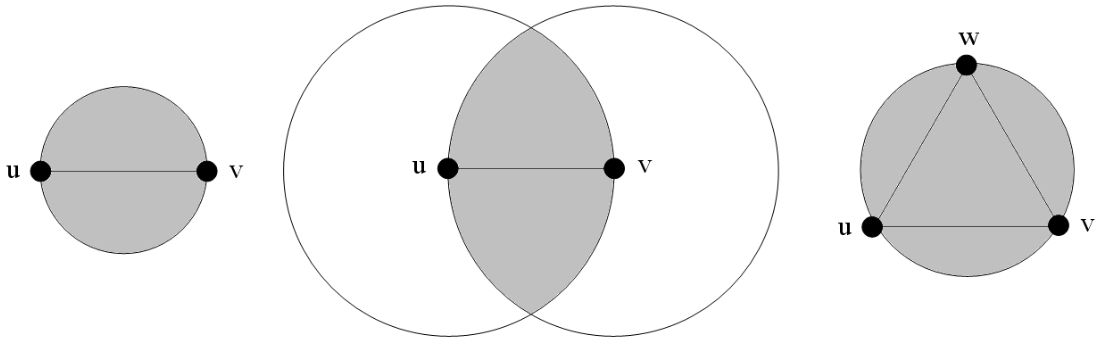

Figure 9: GG, RNG, and Delaunay triangulation.

There are three fundamental problems with these strategies.

Firstly they rely on the assumption that the network can be represented as an idealised unit disk graph model which assumes that nodes are connected if they are within an equal circular transmission range (edges are defined according to an equal threshold represented by the transmission radius). That is, an edge exists between two nodes *u* and *v* (referred to as neighbours) separated by the Euclidean distance *d* if the transmission radius $r \geq d(u, v)$. Secondly, they are sensitive to errors in neighbour location (provided through periodic beaconing). Thirdly there is no informed decision regarding direction of traversal (clockwise or anticlockwise) at a local minimum. This means that the packet may traverse the longer way around the boundary as shown in Figure 10, where a clockwise decision at the local minimum will mean a 7 hop path to the destination and an anticlockwise decision will mean a 20 hop path to the destination.

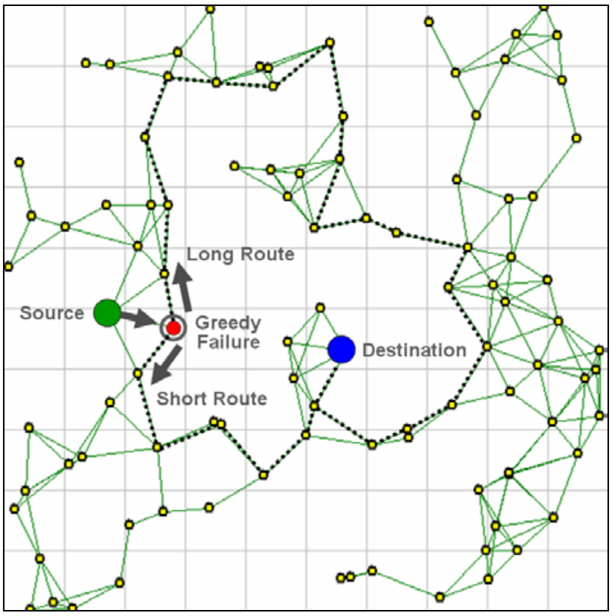

Figure 10: Critical choice required for direction of boundary traversal.

2.5.5 Planar Graph Based Routing Protocols

In this section we will discuss the planar graph based geographic routing protocols in more detail.

Compass II [34] proposed, but did not implement, the idea of extracting a planar subgraph from a unit graph using Delaunay Triangulation. Unlike the following protocols Compass II uses compass forwarding (direction) until a local minimum is encountered and forwarding fails. At a local minimum Compass II suggests that a packet may begin by traversing either edge of the planar subgraph. When an edge is reached that intersects the line from the failure point to the destination, the location is recorded and traversal of the current face continues until the face has been fully traversed. At this point the next face is selected for traversal as the face containing the edge with the intersect point that makes the most progress from the entry point of the current face to the destination.

The concept of face traversal was implemented and tested in Greedy-Forward-Greedy (GFG) incorporates GEDIR [4] as its basic geographic forwarding strategy and FACE-2 [42] (planar graph traversal) to recover when Greedy forwarding encounters a local minimum. The FACE-2 algorithm extracts a planar subgraph in the form of a GG. On Greedy failure a packet will traverse the first face using the right hand rule and will change to the adjacent face at the first intersect of the line from the failure point to the destination if that point is closer to the destination than the point where the current face was entered, unlike Compass II which traverses the entire face and looks for the best face. The GFG-sooner-back algorithm (GFG-s) [6] proposed improvements to reduce the hop counts in GFG. Firstly FACE-2 is modified by introducing two hop neighbour information to determine if there is a closer node to will allow the packet to exit FACE mode earlier than in the previous FACE-2. Secondly GFG-s proposed a shortcut procedure involving 2-hop neighbour information to check for a shorter path than that provided by the immediate neighbour. The number of hops is further reduced by using the dominant set to minimise the nodes involved in route determination.

Greedy Perimeter State Routing (GPSR) [7] is a similar approach to GFG which was implemented as a packet switched routing protocol. Nodes maintain one-hop neighbour location information that is exchanged using periodic beacons. Packets are first transmitted with a mode flag set to Greedy. When a local minimum is reached the flag is changed to perimeter mode and a face traversal algorithm is used until Greedy forwarding can be resumed. GPSR uses the RNG algorithm to construct a planar subgraph. At a local minimum the packet is forwarded to the closest anticlockwise neighbour using the left hand rule (although this is arbitrary as long as it is consistent). When an edge is encountered that intersects the line from the failure point to the destination the next face is traversed (again using the left hand rule). Packets are not permitted to traverse an edge previously traversed to ensure that the packet does not loop. This approach which forces face change has been shown in [43] and [44] to produce routing failures.

An alternate approach has been proposed to extend the face algorithm proposed by [34] although using Greedy forwarding rather than Compass forwarding. Adaptive Face Routing (AFR) [8] extends FACE with Bounded Face Routing (BFR) which places a bound on face traversal defined by an ellipse with the foci at the source and destination. The size of the bound is initially estimated, and then if BFR fails and the packet returns to the source the bound is increased and the BFR process is repeated to a threshold after which the search is terminated (and so the entire face may not be traversed). This attempts to address the problem of uninformed decisions on boundary traversal (and excessive paths for the wrong choice). However the complete boundary may not be searched, additional bandwidth is consumed by failed searches when the search area must be expanded and paths retraced then repeated, and because searches are reactive latency will be increased. AFR is extended in Greedy Other Adaptive Face Routing (GOAFR+) [45], [46]. GOAFR uses a dynamically increasing bounding circle centered at the destination (similar to AFR) on Greedy failure and optimises Greedy fallback by keeping two counters which count the number of nodes traversed in face mode which are closer and further from the local minimum failure point. When set criteria are met the packet is permitted to fall back to Greedy mode even though the packet may not be closer to the destination than the failure point. This provides a more efficient solution than AFR but still suffers from the same limitations. The progressive and unnecessary expansion of the boundary circle is partially addressed in Greedy Other Adaptive Face Routing Adaptive Boundary Circle (GOAFR Plus-ABC) [47] where the boundary is set relative to the distance from the destination to a neighbouring node furthest from the destination.

Greedy Path Vector Face Routing (GPVFR) [48] proposes that nodes on the planar graph build path vectors for each adjacent face (max 3 hops from each node) by exchanging information in beacons (with only a small increase in beacon size). Routing then uses a tri-modal approach with Greedy forwarding, vector face routing, and face routing as in GPSR. On Greedy failure, path vector face routing uses a greedy algorithm to forward packets to a forward anchor node (up to 4 hops away) along the planar subgraph using the accumulated face vector information. On vector face routing failure, face routing as in GPSR is employed. Although GPVFR attempts to maintain some limited global knowledge it is done through the slow beaconing process and so convergence is an issue. Because of this GPVFR is proposed for quasi static networks (and does not perform in heterogeneous environments). Unrealistically it is suggested that beacon intervals can be reduced to reduce convergence but that will increase congestion and contention.

The lack of robustness of the unit graph has been partially addressed in [41] and [49] where a quasi-disk planar graph is proposed to accommodate variations in transmission ranges of 1 to $\sqrt{2}$ (approximately 40%). The graph is defined by two radii r and R where $r \leq R$ and $R / r < \sqrt{2}$. The approach in [49] uses a preceding phase (called the completion phase) to establish virtual links between disconnected nodes before extracting the planar subgraph. This approach is refined in [41] to remove unnecessary virtual links to improve efficiency. This does have a small period of local convergence during which local forwarding decisions may be inaccurate; however, there is no global convergence required.

The Cross-Link Detection Protocol (CLDP) [50] offers a face routing solution for arbitrary graphs (real world heterogeneous networks) by trading off increased bandwidth and increased convergence to improve robustness to radio transmission range variations and coverage and to deal with errors in neighbour location information. This is achieved through the use of probing of local links using the right hand rule to eliminate unwanted link crossovers that would otherwise violate the assumption of a unit graph. The process involves two phases, a prepare phase and commit phase after which edges are tagged as dormant or non-routable. Between phases the edges are locked and probe packets may be lost forcing retransmissions. As all edges must be probed multiple times the probing overhead grows as network density increases.

This approach was tested in a limited fashion in an internal environment with statically placed sensor devices so it is difficult to generalise results and make any evaluations. It was tested in more detail in [51] (in the following paragraph) where it was found that CLDP produced considerable control overhead as expected, with nodes in networks of degree between 6 and 14 sending around 1,500 messages per node.

An alternate approach for heterogeneous networks which attempts to reduce the complexity of CLDP and demonstrates better path selection around local minima is Greedy Distributed Spanning Tree Routing (GDTSR) [51]. For robustness GDSTR builds two spanning trees from either side of the network (rooted at X*min* and

X*max*). To tailor the tree to geographic network each node represents a convex hull containing all descendant nodes and membership is determined by distance from the root. The tree will span around voids allowing informed decisions for direction of boundary traversal unlike the face routing approaches. To build the tree, additional information is propagated through the network from the root within the keep alive (hello) messages inherent to a beaconing based routing system (as is the case with all the planar based face routing protocols in this section). The problem is that it increases the beacon size and takes 3*D* (where *D* is the network diameter in number of nodes) for the tree to converge. On failure it will take a set number of missed beacons to detect a link failure and 3*DT* (where *T* is the average beacon interval for re-convergence) which is excessive (although local maintenance may be possible). A 10 hop diameter with 1.5 second average beaconing interval and 3 missed keep alives for link failure would require 19.5 seconds for re-convergence. For larger networks this would be quite unsuitable. They suggest immediate sending of beacon messages on link failure but this will flood the network while it re-converges and produce contention and congestion problems.

Because the authors claim convergence can be reduced for both GDTSR and CLDP because of available bandwidth (to reduce probe/beacon intervals on link failure) they failed to measure convergence time which is a huge potential problem. In addition the fail to consider bandwidth is required for destination node location distribution and further diminish the effect of this problem by testing dynamicity by removing nodes rather than implementing node mobility.

The planar graph based protocols offer localised low bandwidth solutions with minimal bandwidth, memory and computational requirements as the distributed algorithms to extract a planar subgraph require little overhead. The main problems with this family of protocols is that they are not very tolerant to location error and can not function in heterogeneous environments as they rely on the assumption of a equal transmission radius and uniform coverage assumed to equate to a unit graph. However real world wireless networks may have dissimilar devices with differing wireless capabilities but all will most probably be affected by environmental factors making the assumption of equal circular radio coverage unrealistic [52]. A partial solution to this problem is offered in the quasi-disk planar approach in [41] and [49]. And the only complete solution is CLDP which has only been tested in a limited static environment so few assumptions can be made regarding performance over a realistic range of network topologies, densities, distributions and with mobility.

Another serious problem with this family of protocols (and any other protocol using local knowledge only) is the selection of the wrong direction for boundary traversal at a local minimum. GOAFR+ partially addresses this with bidirectional probing and retracing using a weighted count of nodes that are closer and further from the destination but this is only an estimation and will be more problematic on the outer boundary. This protocol also allows repeated searches at increasing radii which overall will add additional latency to the search process and still may not explore the entire boundary.

The spanning tree approach of GDSTR offers an alternative approach to routing in static heterogeneous environments with lower control overhead, better performance (stretch). However this approach like CDLP is suitable for static networks but due to the long convergence time would suffer in dynamic network topologies.

The elimination of edges may concentrate traffic but not excessively and may improve throughput by reducing the number of active transmitting devices which will improve spatial diversity and spread contention.

2.5.6 Proactive Probing of Boundaries

Other approaches attempting to proactively probe void faces include an early approach by [53] who found that looping of boundary probes occurred due to edge crossovers, and routing using boundary state information failed to achieve 100% path completion on static network topologies. Boundary probing using a right-hand neighbourhood discovery protocol is suggested in the Face Aware Routing protocol (FAR) [54]. This strategy is based upon face traversal of a planar graph which has limitations as discussed previously.

[55] proposes an on-demand void discover strategy for local minima on an inner void

boundary. This approach uses the right hand rule to identify the void, and proposes a rerouting mechanism to select a forwarding side and set a detour around the local minimum that initiated the discovery process. This strategy is functional on a small scale for inner voids of a regular shape, but does not take into account more complex scenarios on the outer boundary and situations where the right hand rule fails due to edge crossovers.

2.5.7 Waypoints (Hybrid)

Waypoints are a scalable solution that is robust in the face of network dynamics. However it concentrates data traffic through fixed areas of the network when alternate paths may exist that would diversify the load, bandwidth consumption and power consumption between nodes.

Terminode routing [10], [56] uses GPSR for basic routing but scales routing to large network topologies using a hybrid approach to routing. Terminode Local Routing (TLR) maintains distance vector routing tables within a set radius of a node. Terminode Remote Routing (TRR) uses a set of anchor or waypoints to route packets through the network. Anchors are established through a discovery procedure in conjunction with cached anchors from “friend” nodes that are considered reliable. After the anchor point for a destination has been discovered, the list of anchor point vectors is inserted into the packet header and the packet is forwarded progressively though the list using geographic forwarding. When a node is reached which has a distance vector entry for the destination, local information is used to complete the route.

The waypoint approach may in some instances offer a scalable solution in large segmented networks, however there will be latency and control overhead involved in discovering and maintaining waypoints. The waypoint approach has the serious problem that it concentrates traffic at waypoints when there are alternate paths through the area that will not be in contention. This strategy may be useful to help scale alternate approaches to larger linked network segment topologies.

2.5.8 Zone Based Strategies

The problem of scalability can be addressed by dividing the network into regions or zones where nodes within a region only hold local knowledge and a representative node maintains limited global knowledge to route between regions. This minimises the nodes involved in any search strategy and minimises the knowledge required by individual nodes.

Depth first search with dominant sets [2] to reduce the number of nodes involved in routing and thus reduce the number of hops involved in route determination, routing is restricted to the dominant set until the destination is known to the current node. When route failure occurs the packet backtracks to the previous node, which forwards the packet to the next closest neighbour to the destination. This has both advantages and disadvantages. Minimising nodes involved in routing reduces contention. However, relying on a subset of nodes for routing can be a problem in dynamic environments when nodes responsible for routing drop out or move out of their neighbourhood, or member nodes move between neighbourhoods.

The Scalable Location Update-Based Routing protocol (SLURP) [11] incorporates location management, which divides a geographical area into rectangular regions called home regions. Each node in the network maintains a location table that maps node ID to the corresponding home area ID for all other nodes in the network. Home region locations are obtained by querying the home region, or asking surrounding nodes if they have the location in a large network. For routing, SLURP forwards packets to the center of a home region using MFR without backwards progression. When a node is encountered that is within the destination home region, SLURP checks for a cached route. If no route exists then SLURP uses source routing similar to DSR to discover a route to the destination. MFR is a strange choice as a basic forwarding strategy for reasons discussed in the section on geographic forwarding. The use of flooding for route requests as in DSR consumes bandwidth and suffers from latency involved in the search and the need for route maintenance.

Like the waypoints approach the concept of zones is a mechanism to scale existing approaches. The problem with any zone based approach (whether geographic or non-geographic) is neighbourhood membership and identification and management of backbone or gateway nodes that represent a neighbourhood and redundancy of information held within the gateway nodes for surrounding zones.

2.5.9 Power Management

This category of protocol takes three approaches. Firstly, routing decisions may be made with consideration of remaining power. Secondly, power may be minimised by reduction of transmitter power to a minimum that will still ensure reliable transmission across each individual connection. Thirdly, nodes within the network may elect a subset of nodes to remain active while the other nodes go into sleep mode to minimise power requirements (and cycle between sleep and wake states).

2.5.9.1 Power as Cost

[57] proposes separating the route calculation to include the link *cost* (a generalised metric) involving factors such as power consumption required for the transmission and link quality parameters such as link delay and packet loss. A combined metric called Normalised ADvance (NADV) is proposed which is the standard forwarding *metric* divided by the newly defined *cost* based on distance to destination and link cost which is based on the packet error rate for the link. The goal of this approach is to increase the packet delivery rate and thereby reduce the power consumption in noisy environments which are prone to packet loss. The cost parameters for the model are adaptable and the options of packet error rate, link delay and power consumption are evaluated for use in link cost estimations. Results show performance improvements using NADV but future considerations of a balanced cost metric are proposed for varying environments. This protocol may be applied as an extension for any geographic forwarding strategy, however its implementation requires an extension to the MAC layer (called the Wireless Integration Sublayer Extension) which limits its application.

Depth first search with dominant sets [2] proposed that the availability of a path may be determined according to alternate metrics representing QoS requirements such as bandwidth and power availability which seeks to make the best use of power resources. Similarly [58] considers power and cost metrics for the next hop selection.

An alternate approach is the partial spanning tree proposed by [59], which builds a spanning tree between self organising clusters based on average residual energy and available bandwidth. These factors are combined into a single measure of difference between clusters called bandwidth-energy product. The proposed distributed algorithm provides a detour tree which does not need to be recalculated when the network topology varies.

2.5.9.2 Power Adjustment

Proactively the Nearest with Forward Progress (NFP) [60] strategy proposed forwarding a packet to the closest neighbour in the forward direction (closest to itself not the destination), then modifying its transmit power to suit the connection. This results in higher delivery rates due to reduced interference and contention at the cost of increased hop count and conserves power.

Geographic Power Efficient Routing (GPER) [61] makes decisions based on power consumption for an event rather than residual power remaining. If GPER determines that the power required to transmit to the next hop candidate requires greater energy than transmitting via a common intermediate neighbour then the intermediate neighbour is selected as the next hop. At this intermediate node the best next hop candidate is determined in the same manner. The IEEE 802.11 standard includes power adjustment options so this type of approach may be practical to implement as an extension to any forwarding algorithm.

Power Boosting Geographic Routing (PBGR) [62] uses a link lifetime estimation to adjust the transmit power (radio range) at a local minimum. PBGR makes decisions of signal strength based on distance (neglecting factors such as transmitter type, antenna gain and atmospheric conditions). The estimated transmit power is compared to a threshold for reliable communications. PBGR takes into account node position and velocity for these calculations and produces a final estimation for expected link lifetime as a moderator for forwarding decisions when it is estimated that a candidate next hop node will move out of range. When a local minimum is detected PBGR proposes a temporary power boost to increase the available neigh options for next hop selection which will reduce the instances where a fallback strategy is required for routing around local minima (in this case perimeter mode of GPSR).

2.5.9.3 Power Management Sleep Mode

STEM [63] proposes a system to minimise power consumption by allowing nodes to rotate through a sleep state using localised scheduling. It is proposed that nodes sleep and then wake periodically to listen for communication requests from adjacent nodes. When a neighbouring node wants to forward packets to a sleeping node it sends out periodic beacons with the recipients ID (on a separate frequency/channel to data traffic) to poll the target neighbour (until it wakes). Because the recipient is guaranteed to wake within a specific time it will respond to the communication request and re-establish the link for data transfer. This has the adverse effect of increasing propagation delays as the sender waits for the target neighbour to wake up.

Span [64] uses a similar scheduling scheme as STEM and ensures that there is a minimum connected backbone for routing (with minimal degradation in network performance). Nodes make a decision whether to sleep based on the number of nodes which will benefit from their decision. Improvements are based on the ratio of routing to sleep time which increases with network density.

The Geographic Adaptive Fidelity (GAF) [65] also proposes a mechanism which allows nodes to sleep. GAF divides a network into a grid with fixed zones. In each zone one node is elected as the cluster head to stay awake for set periods and manage the data transfers within that region. This allows the other nodes within the region to change to sleep mode to minimise power consumption. Nodes wake when they have data to send and they forward it to the cluster head within that cell for routing. The cluster head then forwards the data from one zone to the next towards the base station. The problem with GAF is that it is not heterogeneous, in that it divides the network area into squares based on a percentage of the nominal radio range and assumes all nodes (and links) are uniformly capable of performing at this distance. Nodes locate themselves within a cell based in their location under the assumption that their transmission range is sufficient to cover the cell area and communicate with adjacent cells (but as its conservatively estimated it may actually mean more hops are required for a path). This allows some cells to sleep but may incur additional hops (and therefore power consumption and increased latency).

Geographic Random Forwarding (GeRaf) [66], [67] is a stateless protocol in which nodes cycle though random sleep and wake states creating a random network topology. The cycles are very short (as there are no negotiations required) and as a result the latency will be small in comparison to other competing approaches. When a node has data to send it broadcasts the packet (rather than addressing it to a specific next hop candidate). All listening nodes will receive the packet. The first node closer to the destination to respond will rebroadcast the packet at which time any other node considering transmitting the packet will drop the packet. With this system the cycle rate of nodes in an area can be adjusted so that on average only one is on at a time to minimise contention. In [67] a multichannel MAC layer architecture is described to deal more effectively with contention among potential forwarding nodes according to a priority value based their distance from the destination. The system incorporates two radio channels (on separate frequencies), one for the wakeup signaling (and contention handling) and one for data. Note that GeRaf assumes uniform coverage and as such is not heterogeneous (although alternative coverage models are being investigated).

The power management strategies discussed in this section could be considered generally to improve (extend) other geographic routing protocols, in the case where they do not require customised interaction with custom hardware or alternate MAC layer implementations and operate in heterogeneous environments.

2.5.10 Stateless (Beaconless) Geographic Routing

All of the proceeding geographic routing protocols (other than geographic aided routing) require a neighbour table containing accurate information regarding the status of links (up/down, uni/bidirectional) and accurate location for all directly connected neighbours. This requires the periodic exchange of beacons (or hello messages); however this consumes bandwidth and power and requires for compromise between accuracy, energy consumption, and freedom to sleep. Stateless protocols fall under the heading of power management but represent a category of geographic routing protocol of their own.

An alternate approach which seeks to address this issue is stateless geographic forwarding which eliminates beaconing and adjacency (neighbour) table maintenance. The following section outlines this approach and the inherent problem of dealing with local minima without the availability of neighbour information.

BeaconLess Routing (BLR) [68] first suggested a system in which nodes do not maintain information regarding neighbouring nodes. In the basic forwarding mode a node broadcasts a packet with its location and the destination location using a process termed *volunteer forwarding*. With this approach only receiving nodes that determine that they are within a 60° sector from the previous node to the destination forwarding the packet (under the assumption that they can hear each other). Each of the candidate forwarding nodes delays transmission depending on the progress it makes towards the destination. The closest node to the destination will transmit first as it will have the shortest delay. The other candidate forwarding nodes will detect this transmission and are thereby informed that the packet has been successfully forwarded. The sending node will also hear the transmission and subsequently unicast packets to the self selecting next hop neighbour. Failure at a local minimum is dealt with first by broadcasting a request to the neighbours for their location. All neighbours will respond to this request and the failure node will store the neighbour location information which it will use to construct a localised planar subgraph (GG). The packet is then forwarded through the planar subgraph using the right hand rule in a similar manner to GPSR. When the packet reaches a node closer than the failure point it resumes BLR forwarding.

Implicit Geographic Forwarding (IGF) [69] broadcasts packets expecting all nodes in a 60° sector centered on the track to the destination to consider themselves a potential forwarding nodes in the same manner as BLR. The following is proposed to address possible contention issues and mediate forwarding decisions based on remaining energy levels. When a node has data to send it broadcasts an RTS (modified IEEE 802.11 RTS called an Open RTS - ORTS). Each receiving node within the forwarding area starts a timer based on its distance from the destination and its remaining energy before sending an acknowledgement, in addition to a random delay. Nodes outside this area do not respond. Once a node responds with a CTS all other nodes within its transmission range flag the communication channel as busy. In regards to local minima it presents only a superficial overview of a solution called forwarding shift which retransmits the data and requests a shift (expansion) of the forwarding area to search for a candidate next hop. Details of this process lack a detailed explanation.

Contention-Based Forwarding (CBF) [70] addresses the problem of restricted forwarding area (in which all nodes can hear all other nodes) and also contention management. In the previous protocols the limited forwarding area (a 60° sector) means that potential next hop candidates outside this area that make forward progress towards the destination will be excluded from the negotiation process. Firstly CBF proposes a time based forwarding process similar to those proposed above with the timer value based on distance to destination, energy remaining, and a random factor.

CBF then proposes two further extensions. Firstly to allow all nodes that are closer to the destination to be considered for forwarding, the forwarding area is divided into zones. The central forwarding area is defined by a Reuleaux triangle which has a width equal to the transmission range (in which all nodes can hear all other nodes). Initially only nodes within this area are given the opportunity to respond to a forwarding request. If no node responds within a set period the two zones to the left and right of the central forwarding area (where nodes may exist that are closer to the destination) are in turn given the opportunity to content for forwarding.

A second improvement [70] is proposed that requires consideration of MAC layer capabilities (and therefore non standard hardware implementations). This uses an RTF/CTF scheme (Request To Forward/Clear To Forward). When a node considers itself the best candidate it transmits a RTF back to the sender. The sender replies with a CTF message with the forwarding nodes ID. All other nodes receiving the CTF message with a different node ID cancel their upcoming requests. An active contention resolution mechanism will ensure that multiple packets will not be propagated as may be the case in earlier protocols where acknowledgement for forwarding is not required and nodes do not hear of another instance of forwarding. CBF proposes future work to include consideration of network load and node density can reduce the delay incurred by the contention period. CBF does not propose a recovery scheme at local minima.

Blind Geographic Forwarding (BGF) [71] uses three zones (central and side zones) which are progressively considered for forwarding. On forwarding failure BGF rotates the forwarding area 60° and retransmits the packet into both areas simultaneously (on the assumption that nodes receiving a second copy of the packet will ignore it). To address simultaneous forwarding of packets all nodes start contention timers when they receive the packet. Each packet has the hop count in the packet header. If a node receives a second copy of a packet it will compare the hop count to the stored packet and if they are equal it will not cancel its contention timer. This is called Avoidance of Simultaneous Forwarding (ASF). BGF also investigated forwarding area shapes of sector, circle and Reuleaux triangle. It found that for nodes close to the destination the Reuleaux trangle is a better option, but for nodes further from the destination the sector is better.

GeRaf [66], [67] (discussed previously in the power management section) expands on the collision avoidance and density moderation proposed in CBF. In GeRaf nodes wait according to their distance from the destination (actually according to the band or ring they reside within) within a zone based forwarding area as in CBF. An appropriate contention management solution is proposed using dual channel custom MAC layer as suggested in [67]. However in this article the forwarding approach is proposed independently of [67] and the problem of contention in a stand alone application is problematic as nodes can exist that can not hear each other (hence the reason other approaches limit the forwarding sector to 60°). Using the proposed custom MAC may well provide a viable solution but requiring custom hardware significantly limits its application. GeRaf also makes the assumption that all transmission ranges are equal and uniform which is not the case in real world environments. The problem of routing around local minima is superficially treated.

Priority-based Stateless Geo-Routing (PSGR) [72] tackles the acknowledgement contention issue by considering node density and dynamically formulate forwarding zones. Large zones mean contention issues and small zones mean increased forwarding delay on progressive failures. By tailoring zones according to node density it proposes to minimise acknowledgement delay and contention problems. PSGR uses node density to estimate a forwarding zone size which contains one node only. To deal with non-uniform node distribution and inaccurate estimates of node density PSGR adds a small random delay at each candidate before acknowledging the forwarding request. On forwarding failure a retransmission timer is started and the packet is rebroadcast out into an extended area (to the maximum transmission range) in the hope that a node has moved into the forwarding area. If an acknowledgement is received and the packet forwarded the sending node returns to normal operation. Because this process is not suitable for all situations PSGR suggests broadcasting a bypass probe using the right hand rule to discover a path to a node closer to the destination than the stuck point. Nodes keep a record of probes received to eliminate loops. However the right hand rule has issues in sparse networks with branches, irregular outer boundaries, and unequal transmission ranges with edge crossings.

The beaconless recovery problem is addressed in more detail in the Beaconless Forwarder Planarisation (BFP) scheme [73]. This approach describes dynamic construction of a localised planar subgraph on the fly and to determining the next edge of a planar subgraph traversal using a *Select and Protest* mechanism where possible neighbours in a planar subgraph are determined using a contention process. Neighbouring nodes may then protest incorrect inclusions and correct the subgraph membership. BFP does not specify a proximity graph but considers the GG and RNG.

In the *Selection* phase the forwarding node (initially at a local minimum) broadcasts an RTS signal with its own location and sets a timer to *T*. Each next hop candidate sets its contention timer to *T* times the distance to the forwarder / transmission radius (where transmission radius is a hypothetical value). When the contention timer expires the candidate next hop responds with CTS. All other candidate nodes that receive the CTS and lie within the proximity region (dependent on the proximity graph selected) cancel their contention timers and do not respond. All nodes that respond will do so in order of distance from the sender. In a process called *suppression* a hidden node that receives the response (CTS) but has not received the RTS records the edge as a *violating edge*. In the *Protest* phase hidden nodes start a timer with a value similar to the contention timer and when the timer expires (the closest nodes protest first) they protest against violating edges. The node from which the CTS was sent receives the protest and if it is valid relays this to the sender who removes violating edges from the graph. An alternate planar graph called the Circlunar Neighbourhood Graph (CNG) is also proposed which extends the Gabriel circle to a small degree.

A second recovery protocol called Angular Relaying [73] is also proposed which determines the next hop only (without extracting a planar subgraph). This method uses an angle based delay function to determine next hop candidates. Direction of traversal (left or right hand) can be included if previous node information and direction of traversal is included in the header (otherwise a fixed direction would be required). Then to prevent link crossovers a protest phase is employed in a similar manner to BFP.

Stateless protocols eliminate the need for maintaining a neighbour table which reduces the overhead required for frequent broadcast of beacon packets for monitoring of the link state and accurate updating of neighbour location information. The tradeoff is increased latency for data transfers as nodes delay transmission of data packets based on progress. Contention is a major issue (as it increases latency); however, the solution is zone based forwarding with acknowledgements and handshaking which further increases delays and possibly requires custom MAC layer hardware and software. These protocols also have problems at local minima because there is no neighbour information to make alternate routing decisions. The proposed protocols are limited in viable methods to address this problem. BLR offers a reactive solution which further increases latency and suffers some problems with delivery. BFP is an alternate solution but uses a delay based planar graph extraction which again increases latency but offers the most promising solution. Still, the lack of global information for boundary traversal decisions can mean that packets may choose the longer route around a boundary which can mean excessively long routes especially on the outer boundary.

2.6 Geographic Routing in Sensor Networks

Sensor networks provide a different set of restrictions and facilities to general ad-hoc wireless networks. General ad-hoc wireless networks with production devices use standard MAC layer implementations. Other than power control in IEEE 802.11 there is limited access to MAC layer functionality and the hardware interface.

In contrast, senor networks may be implemented on custom hardware designed in conjunction with a custom routing protocol. This may provide the facility to interact with all levels of the protocol stack, allow for non-standard protocol stack implementations, and provides access to the signals from hardware devices. For this reason routing protocol design for sensor networks has a different set of constraints and support features to those designed for general ad-hoc network devices.

Sensor networks also differ in their primary function which is data collection, aggregation, and data transfer, usually to a central point or base station. If a single aggregation point is used then only a single destination location is required for routing. Unlike an ad-hoc network, a sensor network may require minimal coverage and overlap to ensure connectivity and propagation of data. To ensure the network is not partitioned it has been shown that the maximum usable range must be no more than $\sqrt{2}$ of the nominal transmission range [41], [49]. As coverage (local minima) is a restrictive factor for geographic routing then guaranteed minimal coverage and overlap may reduce local minima.

There are also situations in senor networks where power is not of concern (such as in vehicular systems). However this can not be assumed. Battery technology is also advancing and CPU and memory capability vary between sensor networks and ad-hoc networks. With improvements in memory memorisation may not be a limiting factor.

2.7 Discussion - Opportunities

A considerable portion of research effort has been directed towards planar subgraphs and face traversal. In general this approach is limited as it relies on beaconing, is sensitive to location inaccuracy, and all but one approach performs in fully heterogeneous environments and probes faces to allow informed decisions for direction of face traversal. The problem driving research on this strategy is a philosophical restriction regarding the need to use local knowledge only. However practical routing algorithms need to make informed choices on boundaries and some global knowledge is required to do this effectively.

Reactive approaches to routing around local minima are either slow or incur excessive overhead (flooding), and protocols such as CDLP and GDSTR which are proactive, suffer overhead and convergence problems as they are complex and maintain much more information than is required for effective routing.

A balanced approach to discovering and distributing global knowledge for void and boundary management has not been achieved.

The Beaconless approach offers a promising solution to limiting control overhead but suffers more at local minima than previous approaches due to a lack of knowledge regarding neighbours. PSGR offers an effective geographic forwarding solution. However the best approach for handling local minima is the select and protest mechanism of BFP, but additional delay mechanisms add too much latency and there is no global knowledge for informed decisions at local minima. It would be worthwhile researching the stateless approach of PSGR and attempting to find an appropriate solution to the problem of local minima in the absence of any local or global knowledge.

2.7.1 Important Points from Reviewed Protocols

1. Reduce the incidence of local minima as recovery strategies are expensive
   - Can existing forward strategies be improved?
   - Can forwarding strategies be provided with the freedom to move away from the destination without looping? If so what is the cost.
2. Improve performance using suitable link costs/metrics
   - Link lifetime.
   - Wait for neighbour to move into area (consider direction and velocity).
3. Are there alternatives for acquiring and maintaining boundary information?
   - CDLP and GDSTR probe and maintain information where not required so investigate more efficient strategies.
   - Can probe initiation be reduced to boundary nodes only? How can local minima be better identified?
   - Can a lower granularity representation (cell based represented by occupancy) effectively define the network topology–especially in regards to voids/boundaries–rather than using nodes and edges.
4. Can the problem of local minima for beaconless forwarding be addressed more effectively?
   - What is the minimal way of gathering and representing global information when required (maybe a cell based representation of network occupancy focused on boundary areas only)?
   - What is the cost of making uninformed decisions regarding directional of boundary traversal around local minima (due to a lack of global information)?

2.8 Proposed Solution

For details of the proposed solutions:

1. The improved forwarding strategy is discussed in Chapter 3.
2. The probing and mapping of boundaries is discussed in Chapter 4
3. The proposed boundary state routing protocol is discussed in Chapter 5
4. The proposed occupancy grid is discussed in Chapter 6

2.9 Hypotheses

In respect to these proposals three Hypotheses arise from the research questions.

Hypothesis 1:

A forwarding strategy that provides a greater degree of freedom to move away from the destination without looping will provide a higher rate of path completion than Greedy forwarding.

Hypothesis 2:

A forwarding strategy that provides a higher rate of path completion than Greedy forwarding will improve the performance of an existing geographic routing protocol when used as a replacement for Greedy forwarding.

Hypothesis 3:

A routing protocol using global boundary state information will perform better than an existing fully distributed geographic routing protocol.

Hypothesis 4:

This Hypothesis arose from limitations of the initial boundary mapping protocol:

The use of a low resolution grid occupancy map will reduce the overhead (probe bytes and data structure memory requirements) for global topography mapping of boundaries.

Chapter 3

Geographic Forwarding

The bearings given by the azimuth compass, whilst the ship was aground, were as under:

Dungeness light house,	*SW*
Lidd church,	*W by S½S*
Town of Dim, but taken to be Hythe,	*NW by N*
Cheriton church, then supposed to be Folkstone,	*ENE*
Cliffy eastern extreme of the land, near Dover,	*E½N*

Matthew Flinders
A Voyage To Terra Australis Volume I, Chapter 1, May, 1814

The first problem to be addressed is the enhancement of the basic geographic forwarding strategy to reduce the instances of local minima. For all existing geographic routing protocols, increasing the proportion of hops where the basic geographic forwarding is used in place of a higher overhead secondary strategy (restricted flooding [4], backtracking [5], planar graph conversion using face traversal [6], [7], [8], depth first search [2], [9], and hybrid [10], [11]) will result in an improvement in efficiency of the overall routing protocol.

In addition to this performance improvement there is a necessity to enhance the basic geographic forwarding strategy to route around boundaries containing only obtuse angles to minimise the nodes that will initiate boundary probes and be involved in the mapping process.

In this section we propose an improved composite forwarding strategy called Greedy-BoundedCompass forwarding. This strategy allows a packet to move away from the destination without looping and traverse boundaries where Greedy forwarding would fail. The proposed forwarding strategy Greedy-BoundedCompass uses Greedy forwarding with an improved Compass forwarding called BoundedCompass forwarding as a fallback strategy on Greedy forwarding failure.

3.1 BoundedCompass Forwarding

The earlier discussion of Compass forwarding illustrates the point that preventing a packet from traversing to the previous node is not sufficient to prevent looping. Limiting the selection of the next hop to nodes that are closer to the destination would solve the problem, but this would remove the ability of Compass forwarding to move away from the destination and progress around a boundary where the packet moves away from the destination.

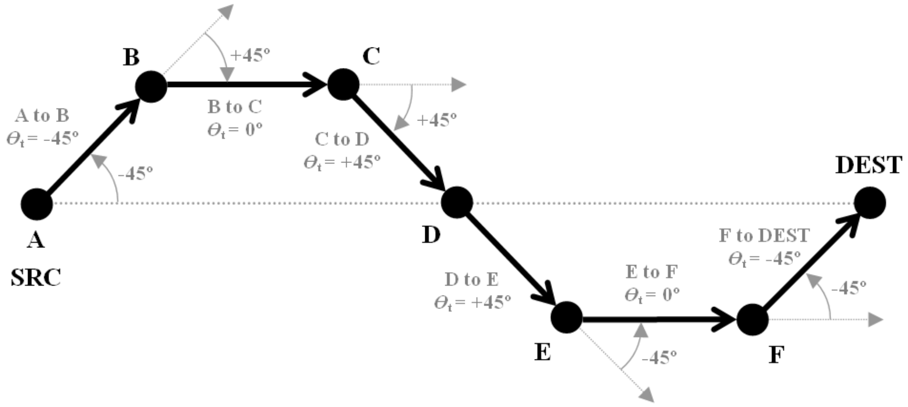

Figure 11: BoundedCompass forwarding using cumulative angle traversed.

Figure 11 demonstrates the concept of maintaining the ***cumulative angle traversed*** from the source (or some other reference point as discussed in the next section) in the packet header and places an upper bound on the angle traversed from the reference point.

The angle traversed Θ_t is initially set to the angle that the first edge traversal deviates from the line between the source and destination. At each node in the path following the initial hop, the (signed) angle change between the previous edge and the next edge is added to the angle traversed.

When selecting the next hop, potential next hop candidates are excluded if the Θ_t would reach or exceed +/-90°. The only exception is the first edge which has no restriction. This will allow a packet to move backwards for one hop at a local minimum (without any additional protocol overhead).

3.1.1 Resetting the Angle Traversed Reference Point

If a packet is forwarded to a node that is closer than any previous node location (the ***Closest Ever*** location) there is no possibility that this next hop selection will result in looping and so no restrictions need to be applied to the next hop selection. At each ***Closest Ever*** location where the next hop must moves backwards, Θ_t will be reset to the angle the next hop deviates from the line from the current node to the destination. Each next hop selection will be restricted to $|\Theta_t| < 90$ until a next hop is available that is closer to the destination than the ***Closest Ever*** location

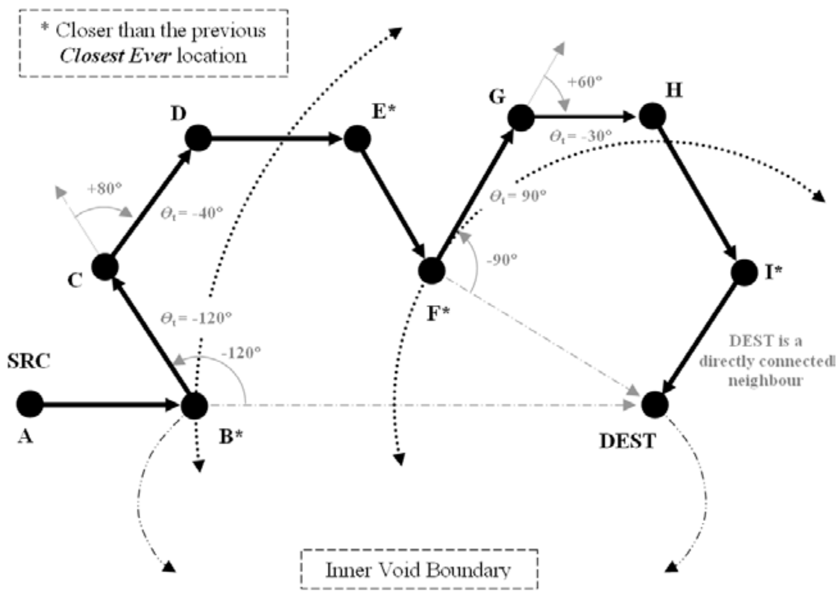

Figure 12: Resetting angle traversed.

Figure 12 illustrates BoundedCompass forwarding in a more complex boundary scenario with resetting of angle traversed at B and F and restriction on angle traversed from C to D and G to H.

The next hop is selected as the node on the closest compass setting to the destination with the following restrictions

If the next hop is closer than the ***Closest Ever*** location (at A, D, E and H):

1. ***Angle Traversed*** Θ_t is not required to be set.
2. The ***Closest Ever*** location is set to the next hop location.

If no node is available that is closer to the destination and the current node is at the ***Closest Ever*** distance (B and F):

1. ***Angle Traversed*** Θ_t is set to the deviation of the edge to the next hop from the line from the current node to the destination.

If no node is available that is closer to the destination and the current node is ***NOT*** at the ***Closest Ever*** distance (C and G):

1. The angle that the next edge deviates from the current edge is added to the ***Angle Traversed*** Θ_t and the next hop is discarded if $|\Theta_t| > 90°$.

At I the destination H exists as a directly connected neighbour and the packet is forwarded to H without any further consideration.

3.2 Combining Greedy and BoundedCompass Forwarding

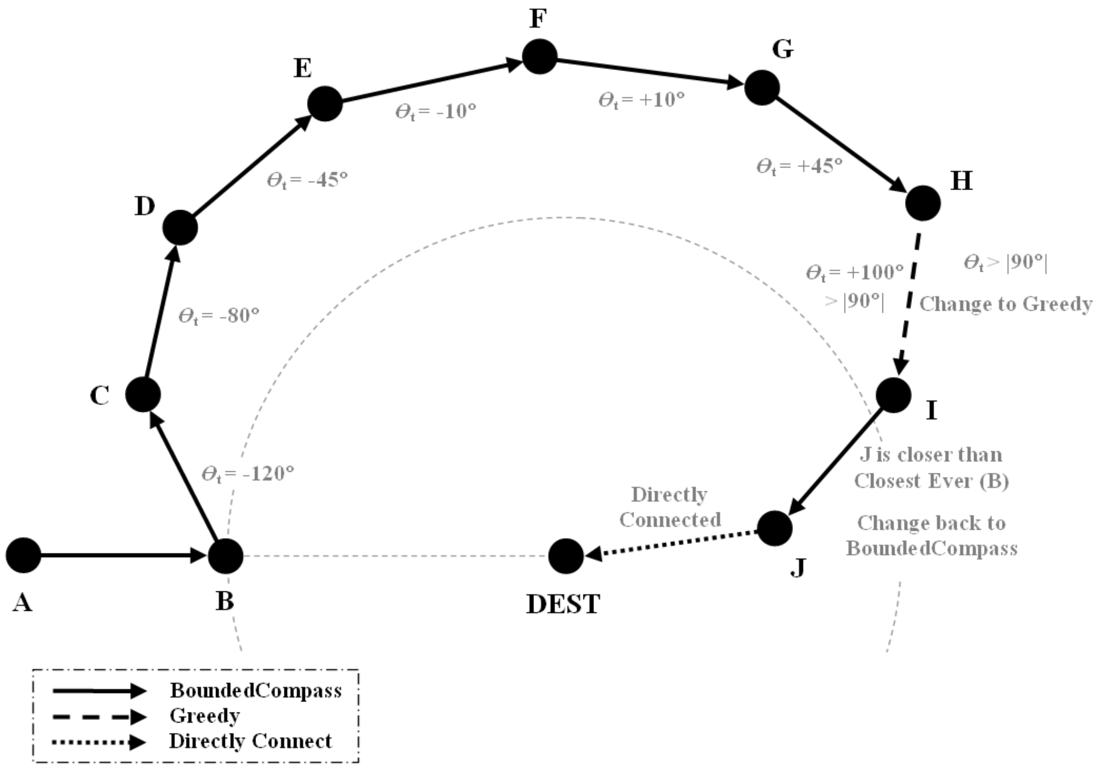

Figure 13: BoundedCompass-Greedy forwarding showing mode changes at H and I.

Figure 13 shows an example where the BoundedCompass forwarding algorithm would fail to reach the destination because at H link traversal to I would result in Θ_t = 100° such that $|\Theta_t| \geq 90°$ and the packet would be dropped. However there is a path available to the destination.

The following section describes two possible algorithms that allow switching between forwarding strategies to address this problem:

1. BoundedCompass-Greedy which uses BoundedCompass forwarding as the primary forwarding strategy and Greedy forwarding as the fallback strategy on BoundedCompass failure,
2. Greedy-BoundedCompass which uses Greedy forwarding as the primary forwarding strategy and BoundedCompass forwarding as the fallback strategy on Greedy failure.

3.2.1 BoundedCompass-Greedy Forwarding

The BoundedCompass-Greedy forwarding strategy begins at the source in BoundedCompass mode as shown in Figure 13.

- At a point where no node exists that is closer to the destination the packet may move to the node on the closest compass heading to the destination without restriction (node B) and the deviation of the edge traversal from the line to the destination is recorded as the angle traversed.
- At nodes following this that remain further from the closest ever location at B (C, D, E, F, G) the next hop is selected as the node on the closest compass heading to the destination with the restriction that the cumulative deviation of the next edge $|\Theta_t|$ is < 90°
- On BoundedCompass failure (H) where no node is available that would produce $|\Theta_t| < 90°$, the packet will change to Greedy mode if a node exists that is closer to the destination (otherwise the packet is dropped).
- The packet will revert to BoundedCompass forwarding (I) only if a node is available that is closer than the closest ever location
- Otherwise the packet will continue on in Greedy mode (and will be dropped on Greedy mode failure).
- Packets may cycle between BoundedCompass and Greedy forwarding as many times as required as long as each transition back from Greedy to BoundedCompass occurs at a next hop that is closer than the Closest Ever location to prevent looping.

3.2.2 Greedy-BoundedCompass Forwarding

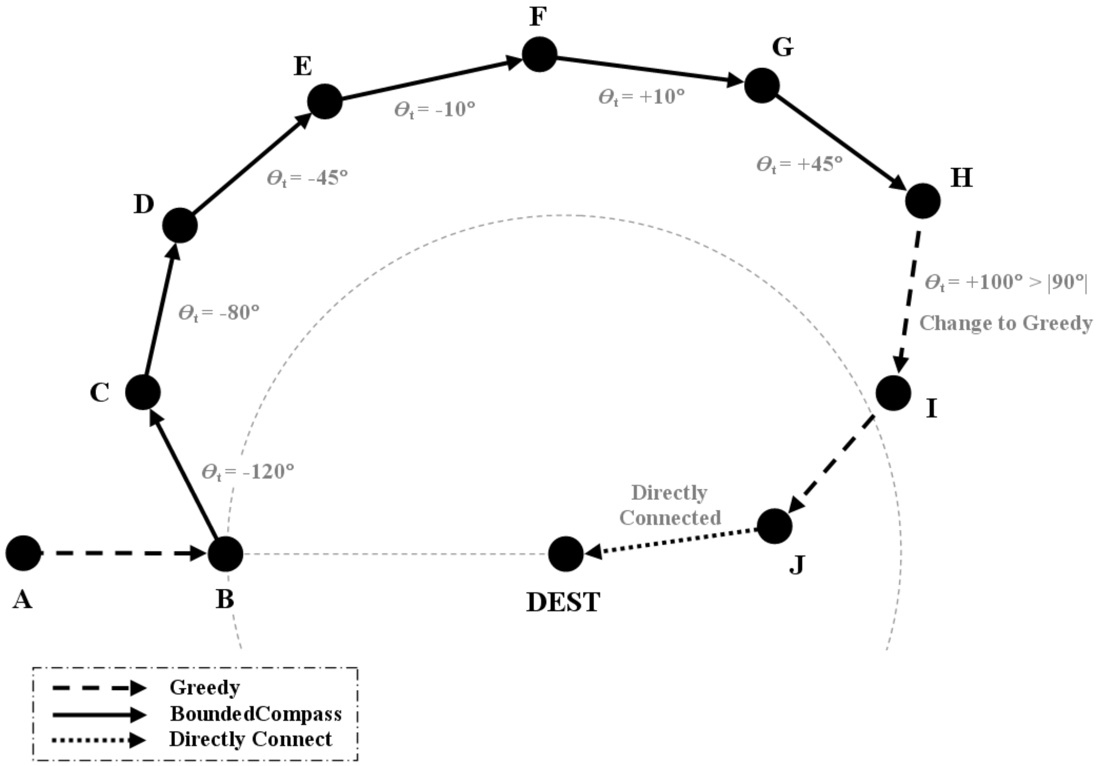

Figure 14: Greedy-BoundedCompass forwarding with mode changes at B, H and J.

Alternatively BoundedCompass forwarding will be tested as a fallback strategy for Greedy forwarding failure as illustrated in Figure 14.

3.2.3 Forwarding Algorithms

1. Pseudo code for the Greedy-BoundedCompass *RouteQuery* algorithm

Algorithm RouteQuery

Inputs: mode, previous node, destination node, previous location
Outputs: next hop, mode, previous location

```
IF current node is source THEN
    mode ← INIT

IF destination is directly connected neighbour THEN
    previous node ← current node
    previous location ← current node location
    mode ← LINK_STATE

    RETURN destination node as next hop

next hop ← Greedy()

IF next hop = FAIL THEN
    next hop ← BoundedCompass()
ELSE
    previous node ← current node
    previous location ← current node location

RETURN next hop
```

2. Pseudo code for the *Greedy* algorithm

Algorithm Greedy

```
Inputs: mode, closest ever location, destination node
Outputs: next hop, mode, closest ever location

next hop = closest neigh by distance to the destination node

IF next hop NOT = FAIL THEN
    IF mode = GREEDY THEN
        closest ever location ← next hop location
    ELSE IF next hop is closer to destination than closest ever location THEN
        closest ever location ← next hop location
        mode ← GREEDY
    ELSE
        next hop ← FAIL

RETURN next hop
```

3. Pseudo code for the *BoundedCompass* algorithm

Algorithm BoundedCompass

```
Inputs: mode, closest ever location, angle traversed, destination node
Outputs: next hop, mode, closest ever location, angle traversed

IF mode NOT = COMPASS THEN
    next hop ← closest neigh by angle
    angle traversed ← angle to destination node - angle to next hop          /* initialise */
ELSE
    next hop ← closest neigh by angle which maintains angle traversed within +/- 90°
    angle traversed ← angle traversed + angle change to next hop

IF next hop NOT = FAIL THEN
    IF next hop is closer to destination than closest ever location THEN
        angle traversed ← angle to destination node - angle to next hop      /* reinitialise */
        closest ever location ← next hop location
    ELSE IF mode NOT = COMPASS THEN
        angle traversed ← angle to destination node - angle to next hop      /* reinitialise */

    mode ← COMPASS

RETURN next hop
```

3.2.4 Data Packet Header Format

The packet format used for the protocol implementation is shown in Figure 15. The previous node location is included to allow updating of the entry in the neighbour table to ensure the packet does not loop due to inaccurate location information.

Dest X, Y	Closest X,Y	Prev X, Y	Prev ID	Angle	Mode
4 + 4 bytes	4 + 4 bytes	4 + 4 bytes	4 bytes	2 bytes	1 byte

Figure 15: Data packet header format.

3.3 Method

Static network topologies were used to evaluate the effectiveness and efficiency of each forwarding strategy in relation to path completion and path efficiency. The application and analysis of the mapping and routing algorithms to mobile environments is proposed for future research when the algorithms have been further developed.

Tests were performed using 100 randomly generated, connected network topologies, each having 100 nodes randomly placed in a 1km square area. Four sets of 100 networks were created with a radio range of 125, 150, 175 and 200 meters with candidate network topologies filtered for a mean connectivity of 2.00, 3.00, 4.00, 5.00+/-0.01 respectively.

For testing data packets were sent between all source/destination pairs in all 100 networks for each forwarding strategy using UDP, providing a total of 9,900 source/destination pairs for each protocol at each radio range for each network.

Because the results were found in many instances to have non-normal distributions the measures are presented as median values with non-parametric statistical comparisons performed using a Wilcoxon Signed Ranks test for paired samples.

Note that for simulation tests in the following section, the presence of a destination location database is assumed. For a more detailed discussion of the experimental design see Chapter 7.

3.4 Comparison of Greedy, MFR and Compass Forwarding

The first section of the testing involved the comparison of the basic forwarding strategies Greedy, MFR and Compass forwarding. All protocols were restricted from traversing back to the previous node to prevent looping. Greedy was further limited to nodes that were closer to the destination as per the basic application of Greedy forwarding in routing protocol applications.

MFR was implemented with no limitation on N (as N relates to radio coverage, power management and interference). Three versions of MFR were implemented: Basic MFR with no restrictions, MFR_FwdProgress which restricted next hop candidates to those making forward progression towards the destination, and MFR_Closer restricted to nodes closer to the destination.

Basic Compass forwarding was implemented without restriction. Compass_Closer was restricted to nodes closer to the destination.

3.4.1 Dependant Variables

The tests were designed to investigate the advantage of additional degrees of freedom provided for both MFR and Compass forwarding in relation to the incidences of looping and the cost of failure for those routes that may have veered away from the optimal trajectory and failed at a local minimum.

Measurements include the percentage of hops where ***looping*** occurred caused routing failure, the percentage of ***path completion***, and ***path efficiency*** measured as the shortest path hops over actual hops. The ***cost of failure*** was measured as the hop count from the source to the failure point plus the shortest path from the failure point to the destination, divided by the shortest path from the source to the destination.

Table I: Percentage of Paths Exhibiting Looping

Strategy	Radio Range/Mean Degree of Connectivity			
	125 meters Conn 2.0 (%)	150 meters Conn 3.0 (%)	175 meters Conn 4.0 (%)	200 meters Conn 5.0 (%)
Greedy	0	0	0	0
Compass	61.3	47.8	35.7	20.2
Compass_Closer	0	0	0	0
MFR	62.2	49.1	35.5	21.7
MFR_FwdProgress	3.0	5.0	4.7	3.3
MFR_Closer	0	0	0	0

Table I shows the percentage of routing loops per network. These results are as expected with the cost (in terms of routing loops) for the freedom to move away from the destination extremely high for Compass and unrestricted MFR. Although much smaller, the incidence of looping is still evident in MFR with forward progress only.

Any capacity for looping limits the applicability for practical use; however, these figures and possible remedies for looping will be addressed later.

From this point on the variable of Radio Range/Mean Degree of Connectivity will be referred to as Mean Degree of Connectivity as this is more generalisable than radio range which is only relevant in relation to the number of nodes and the network size.

3.4.2 Paths Completed

Figure 16 shows that due to the freedom to move backwards at local minima Compass forwarding demonstrated a higher rate of path completion than all other forwarding strategies at all connectivities, with an improvement over Greedy of 18.5% at a connectivity of 2.0, 16% at 3.0, 8.2% at 4.0 and 3.1% at 5.0.

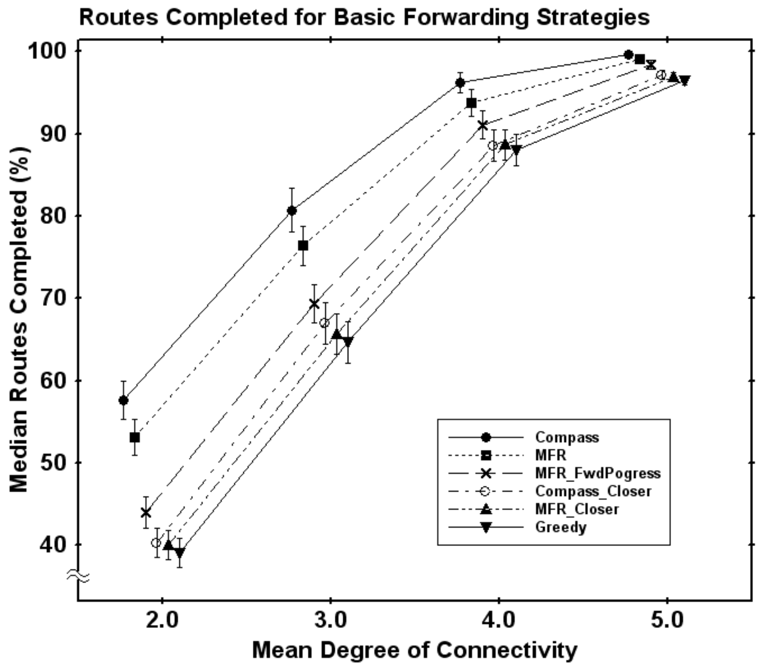

Figure 16: Median percentage of paths completed for the basic forwarding strategies.

This represented a significant improvement over Greedy forwarding as shown in Table II using a Wilcoxon Signed Ranks test.

Table II: Comparison of Routes Completed for Greedy and Compass Forwarding

	Greedy			Compass			Results
Connectivity	M	Mdn	CI	M	Mdn	CI	p
2.0	40.4	39.1	0. 1	58.0	57.6	0. 5	<.001
3.0	64.0	64.7	0. 1	78.6	80.7	0. 5	<.001
4.0	85.5	88.0	0. 1	93.2	96.2	0. 4	<.001
5.0	95.5	96.5	0. 1	98.5	9.96	0. 3	<.001

MFR forwarding also showed a significantly better rate of path completion than all other strategies although not as good as Compass. As discussed previously any improvement here must be considered in relation to the excessive looping shown in Table I. However, these results for path completion illustrate that if the problem of looping can be addressed these strategies may provide a viable alternative to Greedy forwarding.

MFR_FwdProgress, which exhibited lower looping, provided an increase in path completion over the closer constrained strategies, although the path completion is much less than basic MFR and Compass.

3.4.3 Effect on Path Efficiency

Uninformed decisions regarding direction of boundary traversal at a local minimum will mean that in 50% of case the wrong decision will be made and the less direct path chosen.

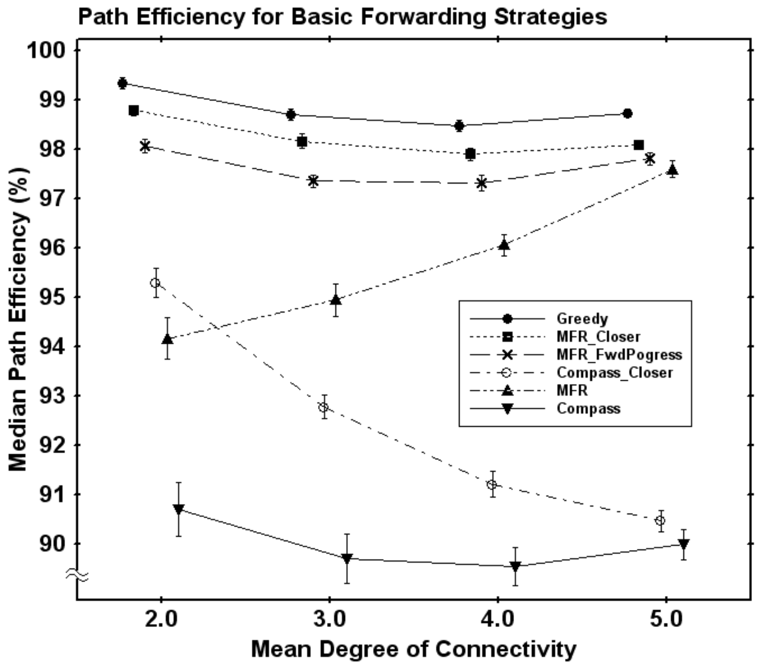

Figure 17: Median of performance for basic forwarding strategies.

This is illustrated in Figure 17 for the measure of path efficiency (for completed routes only). For Compass forwarding an 18.5% increase in path completion over Greedy came at a cost of an 8.5% lower path efficiency than Greedy at a connectivity of 2.0. More important is the lower path efficiency at the higher connectivity with little increase in path completion.

These results must be evaluated with consideration that the degrading of path efficiency may not be as relevant as the improvement in path completion. What is most important is whether looping can be addressed without effecting path completion and how the lower path efficiency impacts on the forwarding strategies application in a routing protocol which may use a far less efficient strategy for routing around local minima.

MFR did not suffer the same degradation in path efficiency as Compass at the higher connectivity. Also its higher rate of path completion came at less cost in terms of path efficiency than the two constrained versions of MFR. MFR_Forward and MFR_Closer were lower in path efficiency than Greedy, but with little improvement in path completion they are of less interest. Compass_Closer demonstrated poor path efficiency in relation to Greedy with a relatively small improvement in path completion.

MFR looks to be worth consideration, but as discussed earlier, MFR may incur additional cost on failed routes. This behaviour will be investigated in the following test.

3.4.4 Cost of Failure

MFR has the capacity to allow packets to veer away from the shortest path even when there are nodes closer to the destination. Packets may therefore fail at a point further from the destination than would otherwise occur, making the potential hop count higher when used in a routing strategy that implements a fallback strategy on failure at a local minimum.

This effect is measured as the route fail dilation, which is the actual hop count from the source to the fail point plus the shortest path hop count from the fail point to the destination, divided by the shortest path hop count from the source to the destination. This represents the best case path expansion that could be achieved if a local minimum recovery strategy was used that could route the packet via the shortest path from the fail point to the destination.

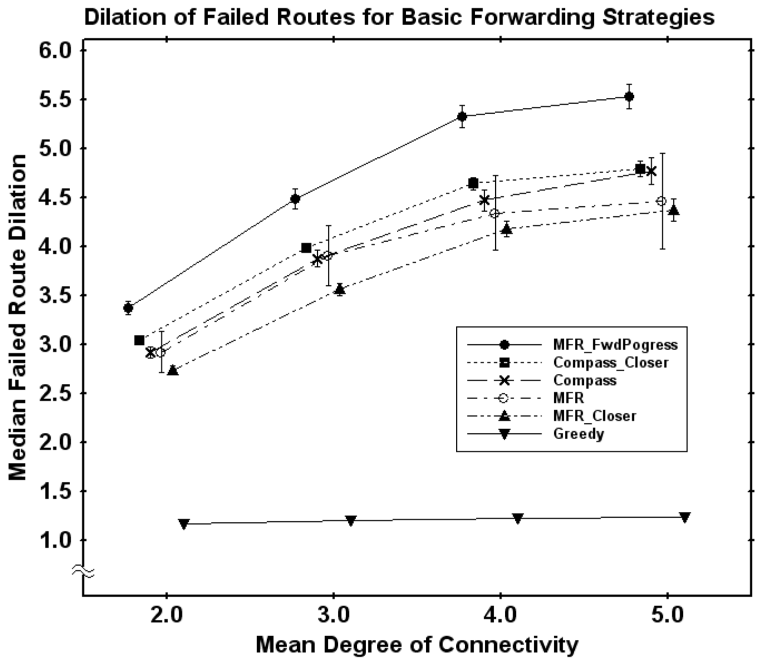

Figure 18: Median fail dilation for basic forwarding strategies.

Both Compass and MFR far exceed the dilation of Greedy indicating that packets are making poor decisions at local minima. For successful routes, Compass forwarding has been shown to have a higher rate of path completion than MFR, while MFR had the better performance. It was expected that MFR would suffer in relation to failed routes while Compass would stay closer to the shortest path. This behavior is evident from the confidence intervals for MFR which shows a wide range of variation in route dilation indicating the effects of both good choices and poor choices at local minima.

However; although Figure 18 shows a slight increase in dilation for MFR in comparison to Compass in the denser network at a connectivity of 5.0, there was no significant difference between Compass and MFR at any connectivity as shown in Table III with comparisons performed using a Wilcoxon Signed Ranks test.

Table III: Route Dilation for Compass and MFR Forwarding

	Compass			MFR			Results
Connectivity	M	Mdn	CI	M	Mdn	CI	p
2.0	2.922	2.923	0.058	2.824	2.927	0.210	0.4990
3.0	3.854	3.886	0.083	3.681	3.917	0.307	0.3772
4.0	4.493	4.478	0.109	4.234	4.350	0.381	0.1769
5.0	4.700	4.776	0.136	4.185	4.470	0.487	0.2378

At this stage of testing Compass forwarding offers the more likely choice as the basis for an improved forwarding strategy, although the MFR option can not be discounted. The next step is to compare the bounded versions of Compass and MFR with the fully constrained versions which are prohibited from moving away from the destination.

3.5 Bounded Traversal with Constrained Angle Traversed

The next set of tests evaluate the performance of both Compass and MFR with the proposed restriction on angle traversed of +/- 90°. For both BoundedMFR and BoundedCompass, results showed neither strategy exhibited any incidents of looping.

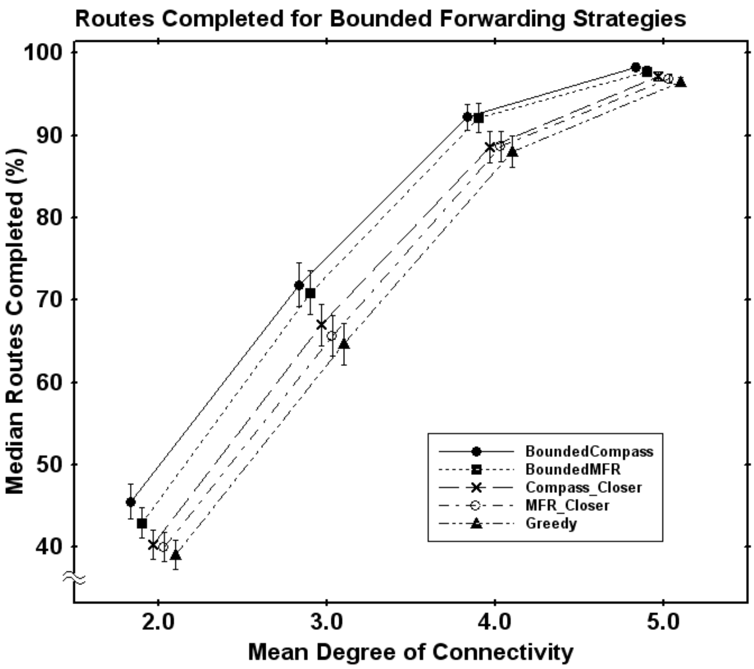

Figure 19: Median of routes completed for bounded forwarding strategies.

In Figure 19 BoundedCompass shows the highest level of path completion although these improvements are far less than those shown by the unrestricted Compass and MFR in Figure 16. These results only represent a significant improvement in routes completed for BoundedCompass over BoundedMFR at a connectivity of 2.0 as shown in Table IV with comparisons performed using a Wilcoxon Signed Ranks test.

Table IV: Routes Completed for BoundedCompass and BoundedMFR Forwarding

	BoundedCompass			BoundedMFR			Results
Connectivity	M	Mdn	CI	M	Mdn	CI	p
2.0	46.2	45.6	2.1	438	430	1.9	< 0.01
3.0	70.8	71.9	2.7	696	709	2.7	0.1667
4.0	89.7	92.2	1.6	889	922	1.8	0.1641
5.0	97.3	98.3	0.5	968	978	0.6	0.1608

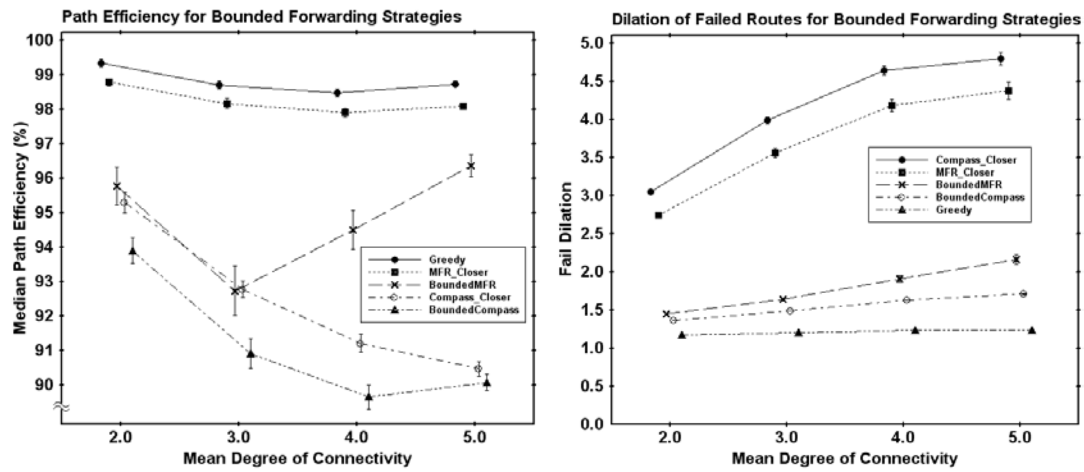

Figure 20: (a) Median performance and (b) fail dilation for bounded forwarding.

For the small improvement in performance of BoundedCompass over BoundedMFR, BoundedCompass incurs a penalty in terms of path efficiency at the higher node densities as shown in Figure 20(a). In contrast Figure 20(b) shows that BoundedCompass had lower fail dilation than BoundedMFR.

In absolute terms these results are relevant to the decision of whether both Compass and MFR continue to be viable options to continue development. Where it was expected that MFR would not perform as well as Compass this has been shown not to be the case.

The results above will be further affected in the next section where multiple strategies are employed with a secondary strategy used as a fallback at local minima, and so further analysis will be deferred.

3.6 Multi-Strategy Forwarding

This section will evaluate the multi-strategy approach proposed as an alternate to Greedy forwarding. The combinations to be compared are detailed below.

1. ***BoundedCompass-Greedy*** forwarding which uses BoundedCompass forwarding with Greedy forwarding on BoundedCompass failure.
2. ***BoundedMFR-Greedy*** forwarding which uses BoundedMFR forwarding with Greedy forwarding on BoundedMFR failure.
3. ***Greed-BoundedCompass*** forwarding which uses Greedy forwarding with BoundedCompass forwarding on Greedy failure.
4. ***Greedy-BoundedMFR*** forwarding which uses Greedy forwarding with BoundedMFR forwarding on Greedy failure.

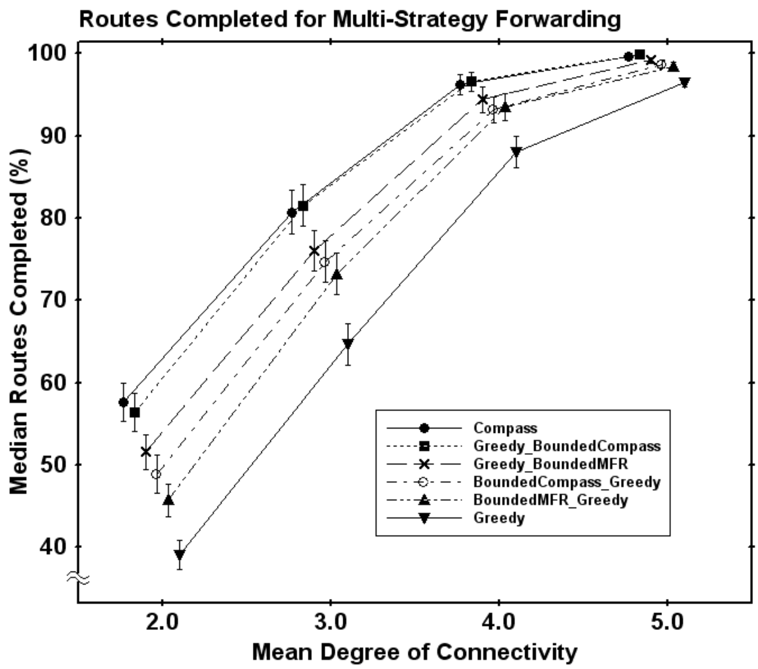

Figure 21: Median of routes completed for multi-strategy forwarding.

In all tests all alternate Compass forwarding strategies were found to be loop free.

Figure 21 shows that Greedy-BoundedCompass demonstrated a significantly higher rate of path completion (as evident from the confidence intervals) over Greedy forwarding, comparable to the basic Compass forwarding, but without demonstrating looping. The improvement was 17.3%, 16.9%, 8.6%, and 3.5% in the median of paths completed at a connectivity of 2.0, 3.0, 4.0 and 5.0 respectively (all significant at $p < .01$ using a Wilcoxon Signed Ranks test).

Table V shows that with a Wilcoxon Signed Ranks test there is no significant difference between Greedy-BoundedCompass and the basic Compass forwarding at a connectivity of 2.0, 3.0, and 4.0. It does show a significant difference at a connectivity of 5.0, however the difference is extremely small (0.4%).

Table V: Routes Completed for Compass and Greedy-BoundedCompass

	Compass			Greedy-BoundedCompass			Results
Connectivity	M	Mdn	CI	M	Mdn	CI	p
2.0	58.0	57.6	2.4	57.2	56.4	2.3	0.055
3.0	78.6	80.7	2.6	79.7	81.6	2.5	0.192
4.0	93.2	96.2	1.2	93.9	96.6	1.2	0.131
5.0	98.5	99.6	0.3	98.9	100.0	0.2	< 0.01

Figure 21 shows that between Greedy-BoundedCompass and Greedy-BoundedMFR, Greedy-BoundedCompass has a small but significantly higher rate of path completion, however Table VI shows that this difference is only significant at a connectivity of 2.0 and 5.0.

Table VI: Routes Completed for Greedy-BoundedCompass and Greedy-BoundedMFR

	Greedy-BoundedCompass			Greedy-BoundedMFR			Results
Connectivity	M	Mdn	CI	M	Mdn	CI	p
2.0	57.2	56.4	2.3	52.2	51.6	2.1	< 0.01
3.0	79.7	81.6	2.5	74.5	76.0	2.5	0.023
4.0	93.9	96.6	1.2	91.7	94.4	1.6	0.012
5.0	98.9	100.0	0.2	98.2	99.2	0.4	< 0.01

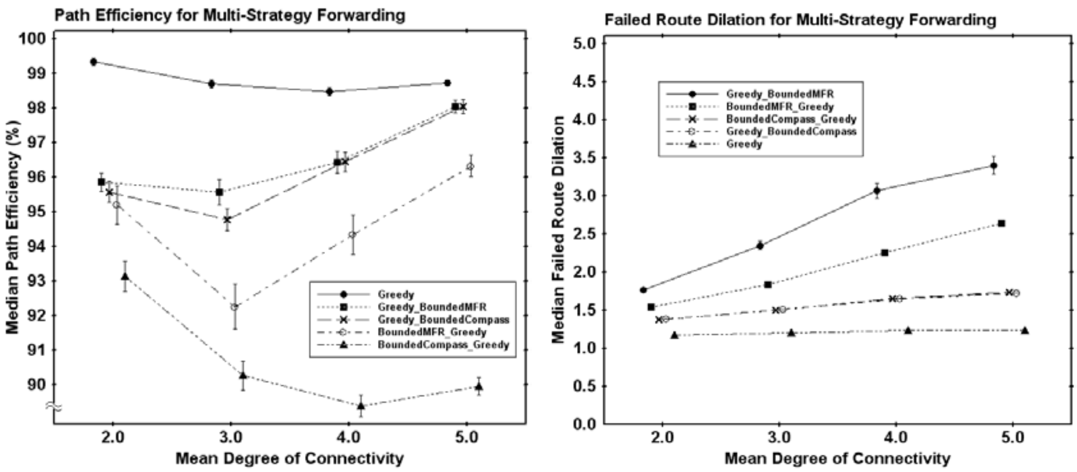

Figure 22: (a) Median path efficiency and (b) fail dilation for multi-strategy forwarding.

In relation to the cost of the higher rates of path completion Greedy-BoundedCompass showed a significantly lower path efficiency than Greedy forwarding. The decrease was 3.7%, 3.9%, 2.0%, and 0.6% in the median of path efficiency at a connectivity of 2.0, 3.0, 4.0 and 5.0 respectively (all significant at $p < .01$ using a Wilcoxon Signed Ranks test).

The cost in regards to failed route dilation for Greedy-BoundedCompass was significantly higher than Greedy forwarding. The increase was 21.2%, 30.4%, 42.3%, and 49.2% in the median of failed route dilation at a connectivity of 2.0, 3.0, 4.0 and 5.0 respectively (all significant at $p < .01$ using a Wilcoxon Signed Ranks test).

Figure 22(a) shows that Greedy-BoundedCompass had a lower path efficiency than Greedy-BoundedMFR. This is shown in Table VII to be significant in the sparse networks using a Wilcoxon Signed Ranks test. In contrast Greedy-BoundedMFR shows a significantly higher failed route dilation than Greedy-BoundedCompass, as clearly evident in the values and confidence intervals in Figure 22(b).

Table VII: Path Efficiency for Greedy-BoundedCompass and Greedy-BoundedMFR

	Greedy-BoundedCompass			Greedy-BoundedMFR			Results
Connectivity	M	Mdn	CI	M	Mdn	CI	p
2.0	95.4	95.6	0.3	95.7	95.9	0.3	< 0.01
3.0	94.7	94.8	0.3	95.3	95.6	0.4	< 0.01
4.0	96.2	96.5	0.3	96.4	96.4	0.3	0.152
5.0	97.7	98.1	0.2	97.7	98.1	0.2	0.293

The results of path completion for Greedy-BoundedCompass must be considered in relation to the cost. In terms of route efficiency for Greedy-BoundedCompass this is a maximum of 1.8% lower than Greedy-BoundedMFR at a connectivity of 2.0, whereas the cost for Greedy-BoundedMFR in terms of failed route dilation is 197% higher.

3.7 Improved Forwarding Strategy

BoundedCompass forwarding was initially conceived for the purpose of forwarding packets around boundaries with obtuse nodes in situations where the packet is required to move away from the destination. It was considered that this would restrict the incidence of local minima to reflex angles only which were easy to identify. Without this facility, boundary mapping would require boundary probes to be initiated from every node resulting in excessive control overhead and making the concept of boundary probing and mapping unviable.

The idea that an improved forwarding strategy was possible had further implications, in that an improved rate of path completion would mean a reduction in the use of fallback strategies for geographic routing protocols which are in general far more inefficient than basic geographic forwarding. It was proposed that Compass forwarding with its cumulative angle traversed limited to within +/-90° (and reset at each node that was closer than the closest ever location) was a viable strategy to improve path completion and address looping. It was initially considered that MFR would not perform well because it could move away from the destination even when nodes existed that were closer to the destination. However results have shown that MFR was a viable alternative all through the testing.

In respect to these proposals it has shown that both the bounded forwarding algorithms and the multi-strategy forwarding strategies performed far better than Greedy forwarding in terms of path completion (without exhibiting looping), with the multi-strategies strategies out performing the bounded strategies.

Greedy-BoundedCompass was considered the better option as an improved forwarding strategy over Greedy-BoundedMFR. Greedy-BoundedCompass was marginally better in path completion. Although its route efficiency was slightly higher (1.8% max), its failed route dilation was lower (51.0% max) indicating that for Greedy-BoundedMFR more failed routes deviated from the optimal path. Still, the close results mean that Greedy-BoundedMFR was a viable alternative to Greedy-BoundedCompass forwarding.

3.8 Application of Greedy-BoundedCompass Forwarding

This section applies Greedy-BoundedCompass forwarding to an existing geographic routing protocol to test the effectiveness of the proposed improved forwarding strategy and to provide a control for later testing with boundary mapping information for routing around local minima.

3.8.1 Choice of GPSR

Greedy Perimeter State Routing (GPSR) was chosen for testing because it is commonly referred to in the literature, uses Greedy forwarding, and more importantly because it employs a fully distributed algorithm which is important for comparisons in later tests relating to the cost and benefits of maintaining some level of global knowledge for use in routing decisions.

3.9 Comparison of GPSR and Improved GPSR

To test the application of Greedy-BoundedCompass as an improved forwarding strategy, it was substituted for Greedy forwarding GPSR.

3.9.1 Modified GPSR Algorithm

The pseudo code below shows the modification to the GPSR *RouteQuery* algorithm to include Greedy-BoundedCompass forwarding.

1. Greedy forwarding is constrained as in Greedy-BoundedCompass such that it can move to a node that is closer than the closest ever location
2. Algorithms for *GpsrGreedy*, *GpsrEnterPerimeterMode* and *GpsrPerimeterMode* are the same as for standard GPSR. GPSR Greedy forwarding is basic greedy forwarding (with the restriction that the packet can not move backwards or move to the previous node, to prevent looping).

Algorithm RouteQuery

```
Inputs: mode
Outputs: next hop

next hop ← Greedy()

IF next hop NOT = FAIL THEN
    mode ← GREEDY

IF next hop = FAIL AND mode NOT = GPSR_PERIMETER_MODE
        AND mode NOT = GPSR_GREEDY_MODE THEN
    next hop ← BoundedCompass()

IF next hop = FAIL THEN
    IF mode NOT = GPSR_PERIMETER_MODE THEN
        next hop ← GpsrGreedy()
        IF next hop = FAIL THEN
            next hop ← GpsrEnterPerimeterMode()
            mode ← PERIMETER_MODE
        ELSE
            mode ← GPSR_GREEDY_MODE
    ELSE IF mode = GPSR_PERIMETER_MODE THEN
        next hop ← GpsrPerimeter()

RETURN next hop
```

3.9.2 Improved GPSR Results

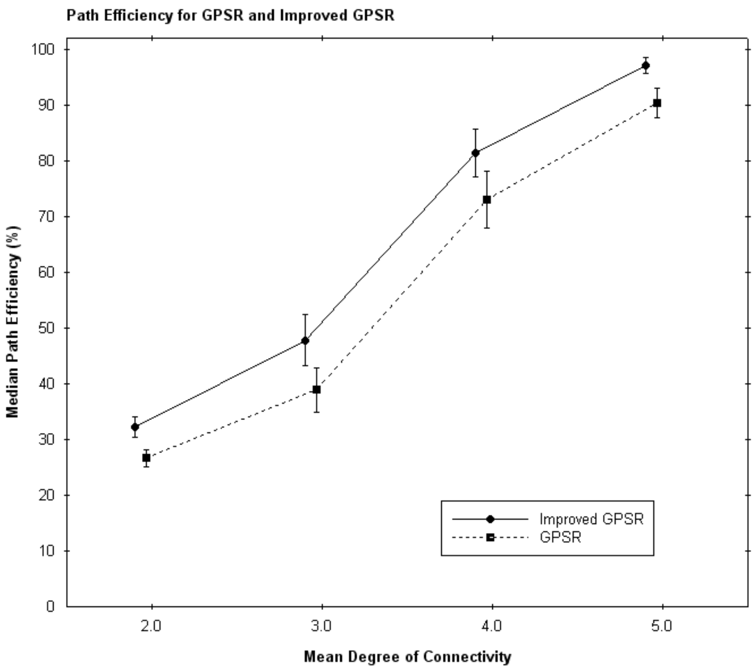

Figure 23: Median path efficiency for GPSR and improved GPSR.

Figure 23 shows that GPSR with Greedy-BoundedCompass performs better at all degrees of connectivity than standard GPSR. Using Wilcoxon Signed Ranks test the improvement in path efficiency was significant at all degrees of connectivity with $p < 0.01$. The improvement in median path length for improved GPSR in comparison to standard GPSR is shown in Table VIII.

Table VIII: Improvement in Median Path Length for Improved GPSR in Comparison to Standard GPSR

	Connectivity			
	2.0	3.0	4.0	5.0
Improvement (%)	5.6	8.9	8.4	6.6

3.10 Discussion

Greedy-BoundedCompass was shown to significantly improve the path efficiency of GPSR in sparse networks when used as a replacement for Greedy forwarding. For example in the sparse network at a connectivity of 2.0 the increase of 17.3% increase in path completion of the improved forwarding strategy Greedy-BoundedCompass with a decrease in path efficiency of 3.7% and an increase in fail dilation of 21.2% in relation to Greedy forwarding, equated to an increase in path efficiency of 5.6% for the improved GPSR with Greedy-BoundedCompass forwarding.

For the application of the improved forwarding strategy to other geographic routing protocols, the improvement in path efficiency will vary depending on the strategy that the routing protocol employs to deal with failures at local minima.

Chapter 4

Boundary Mapping Protocol

I beg you will be particular so far in looking for the track of my party returning, as you will perceive by the map that many very circuitous detours may be thus avoided.

Thomas Mitchell
Letter to Surveyor-General's Office, Sydney, 22d February, 1847

The aim of boundary probing and mapping is to detect nodes within a network where potential local minima exist and map the boundaries (or sections of boundaries) on which these nodes lie, to provide minimal link state information to make an informed choice regarding the most efficient path to route packets around local minima.

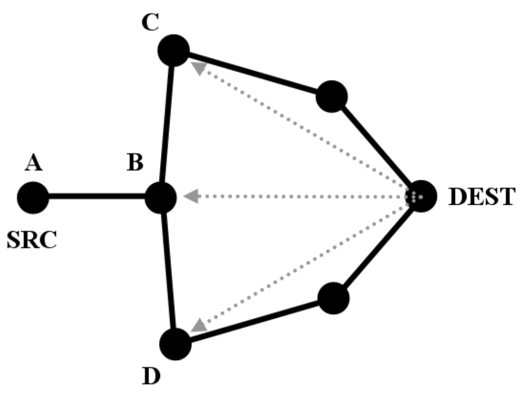

Figure 24: Local minima at a node at an obtuse (not acute) angle.

Figure 24 shows a network where a node (B) having a reflex angle on the void boundary presents a local minimum because its neighbours C and D are both further from the destination than B.

This illustrates that with Greedy forwarding local minima can be difficult to detect.

To probe for boundaries the option in this situation is to transmit discovery probes from all network nodes; however, this is considered too costly, and therefore an unviable option.

As discussed previously if the initiation of boundary probing could be limited to nodes at potential local minima, and local minima could be restricted to nodes at a reflex angle only, then the probing of boundaries could be viable. This influenced the development of the improved forwarding strategy (Greedy-BoundedCompass) which will forward data packets around boundaries with nodes that are not reflex (even if the packet moves away from the destination). Greedy-BoundedCompass has the added advantage that it reduces the overall instance of local minima, thus minimising the need for a recovery strategy on forwarding failure.

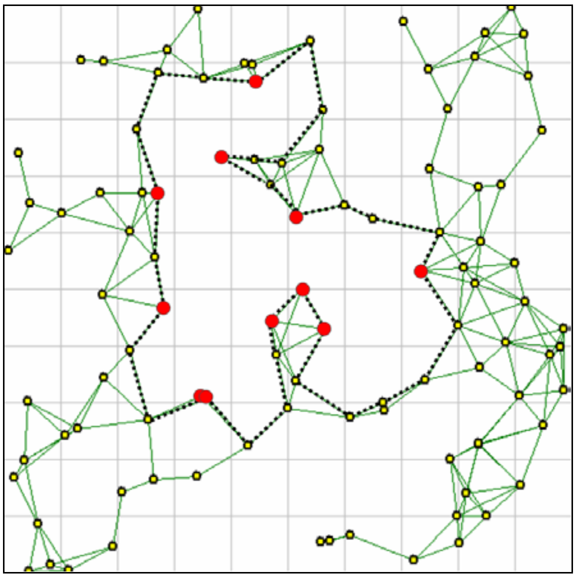

Figure 25: Example of probing from nodes having obtuse angles only.

This allows probing to be limited to potential local minima identified as nodes having a reflex angle only which reduces the number of probe initiations as shown for the internal void boundary in Figure 25.

4.1 Problem Definition

The advantage of geographic forwarding is in the use of distributed algorithms which do not require the maintenance of global knowledge of nodes and links. This basic principle has been extended to geographic routing protocols such as the planar family which includes GPSR. However, incorrect decisions regarding the direction of boundary traversal at a local minimum by protocols with algorithms using local knowledge only can result in excessively long routes when the alternate choice of direction provides a more optimal path to the destination.

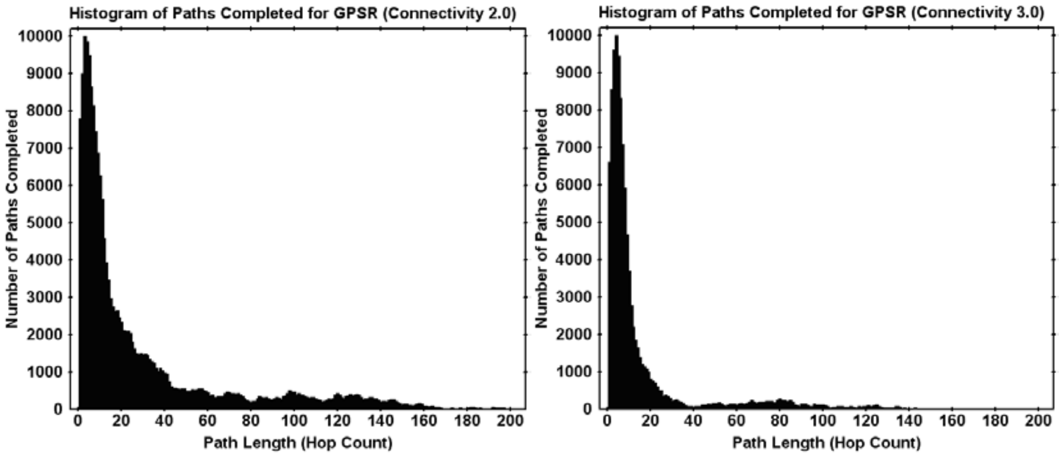

Figure 26: Histogram for paths completed for GPSR in sparse networks.

Figure 26 shows the number of paths at each path length for the distributed geographic routing protocol GPSR in the sparse networks with a connectivity of 2.0 and 3.0 where this problem is most pronounced. The distributions show the relatively large proportion of long routes due to uninformed decisions on the direction of boundary traversal, illustrating that the lack of global knowledge can be a critical factor in routing protocol performance. In the sparse networks with a connectivity of 2.0 the inappropriately long routes actually extend up to 306 hops (requiring the IP TTL to be increased to 512 max to be able to measure the full range of values).

4.2 Proposed Solution: Boundary Probing and Mapping

This chapter will investigate the discovery and maintenance of boundary state mapping information for use in routing packets around local minima.

Problems to Solve

1. Boundary discovery
2. Looping of probe packets
3. Multiple discovery probes for the same boundary
4. Storage of link state information.

Challenges in the probing of network boundaries include minimisation of data header control bits, minimisation of control traffic, and minimisation of the frequency of control traffic.

Note: The illustrations in the following section apply to the *forwarding of boundary probe packets* and *NOT data packets* as in the preceding sections.

4.3 Boundary Mapping Protocol – Simple Version (BMPs)

The following section will describe the simple version of the Boundary Mapping Protocol (BMPs) algorithm for use with unit graphs.

4.3.1 Boundary Discovery

Probe initiation will occur when a node detects that adjacent neighbours are equal to or greater than 180° apart (assuming they are not already adjacent neighbours on an existing boundary). The boundary probe will be forwarded around the boundary using the wall follower rule for traversal of a maze. This rule employs either the left hand or right hand rule, which guarantees that by keeping one hand in contact with one wall of the maze, the exit will be reached if one exists; otherwise, the person will return to the entrance (assuming that the maze is simply connected). For this implementation and discussion we will use the right hand rule and forward to the next clockwise neighbour from the node from which the probe packet was received.

This will result in a clockwise traversal around the outer boundary and an anticlockwise traversal around inner boundaries.

Whenever a node initiates a boundary probe it creates a boundary record, sets its status to ***Discovery,*** generates a locally unique boundary ID and inserts the boundary

ID into the record. The combination of source node ID (generated from an IP or MAC address) and boundary ID will uniquely identify the probe for all boundary member nodes. The source node then sends a ***Discovery*** probe containing its source node ID and the boundary ID number to the clockwise neighbour.

When a node receives a discovery probe from a neighbour it creates a boundary record and sets its status to ***Discovery***. The node stores the node ID of the previous boundary node that forwarded the probe, so that it can detect subsequent discovery probes for the same boundary. Any boundary probe received from that neighbour is considered to represent the same boundary. A node may be a member of a number of boundaries. If a node belongs to adjacent boundaries, each probe will be received from a different neighbour and will therefore be identified as belonging to different boundaries. That is, the identifying feature of each boundary is the left hand neighbour from which the probe packet enters. As probe packets all traverse in the same direction no two probe packets can enter over the same link if they are traversing different boundaries.

If a subsequent probe is received from a neighbour that is already the previous neighbour for a boundary record in the discovery phase, then the following handles this overlap. If the probe source node ID of the discovery probe has a lower source node ID than that in the boundary record, then the record is updated with the new probe source node ID and boundary ID and the packet is forwarded, otherwise the discovery probe is dropped. This process ensures that only one probe (with the lower source node ID) will traverse back to its source.

4.3.2 Branches

A boundary may have offshoots or branches. Rather than treating these as separate entities, branches are included in the boundary to which they are attached. The boundary routing protocol can eliminate traversal of these branches later in the route determination process.

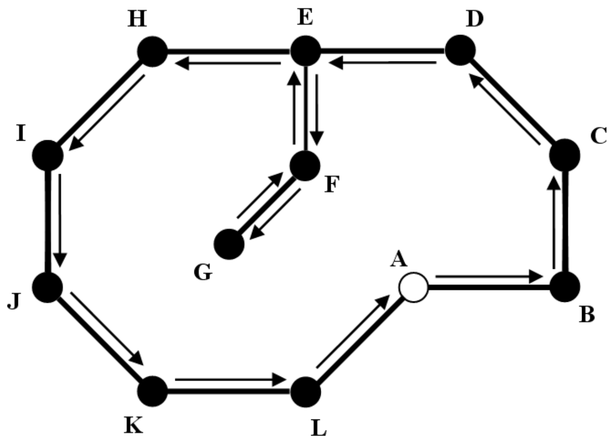

Figure 27: Internal boundary with branch.

Figure 27 shows an internal boundary with a branch (E, F, G). A probe initiated at A will traverse the branch as shown.

4.3.3 Edge Crossovers

The problem identified in previous research was the looping of packets in topologies where perimeter vertices are not coincident with nodes (boundary is not simply connected). In such situations the crossover of edges will cause the simple algorithm described above to fail and may produce looping of the probe packet.

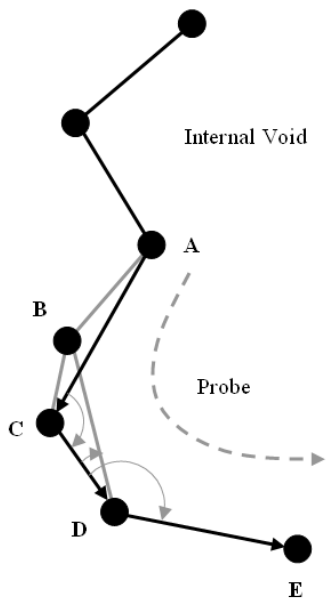

Figure 28: Example of a boundary edge crossover.

This problem of boundary edge crossovers is illustrated in Figure 28. Without additional knowledge the problem can be addressed by eliminating a next hop candidate if traversal to that node would cross a previously traversed edge. This requires that the traversed node list and there locations are maintained in the probe list. Without this mechanism the probe would be initiated from A to C and then to D. At D the next clockwise neighbour from C is B which would direct the probe towards the source. However at D with B excluded because it crosses the edge A, C the next clockwise neigh E is selected and the probe is forwarded from D to E.

The simple probe forwarding algorithm is presented below.

Algorithm ForwardProbe

```
Inputs: candidate next hop, probe list        /* probe list of node IDs */
Outputs: next hop

/* Starting with link to node that sent probe (assumes neigh list is ordered by angle) */

FOR each candidate node on RH rotation DO
    FOR each edge in probe list DO
        IF edge to candidate next hop crosses edge in list THEN
            CONTINUE
RETURN candidate as next hop
```

4.3.4 Detecting Probe Home (Back at Source)

A probe will be assumed to be home if the incoming edge was the left hand edge of the local minimum where it was initiated, or the outgoing edge calculated for the next hop is the right hand edge of the local minimum.

4.3.5 Inner/Outer Boundary

When the full boundary polygon has been traversed and the packet returns to the originating node, the initial probing phase is complete. The originating node then analyses the probe coordinate list and computes the total angle change of the path. If this equals 360° then an interior void has been found. Otherwise if -360° is obtained then the outer network boundary has been traversed.

4.3.6 Distribution of Mapping Information

When the probe packet arrives back at the source a boundary ***Confirmation*** packet will be sent back around the accumulated path list to confirm to each node that it is a member of the identified boundary. At each node the boundary node list is copied into the boundary record matching the boundary ID and made available for routing decisions.

In the next section we will describe an enhanced version of the Boundary Mapping Protocol (BMP) that uses a number of heuristics to deal with problems that arise with non-uniform radio range.

4.4 Improved BMP – Dealing with Non-Uniform Radio Ranges

The approach outlined above would work for networks topologies that are equivalent to a unit graph where all radio ranges are equal. However real world wireless environments have non-uniform radio coverage where some lower powered nodes may not be able to establish bidirectional links with other nodes with a higher power. In this situation the proposed algorithm may fail due to more complex link crossovers where the simple link crossover detection algorithm would not work.

4.4.1 Algorithms for Handling Non-Uniform Radio Ranges

Consider a simple edge crossover on a boundary as illustrated in Figure 29. This configuration can be handled by the simple edge crossover detection algorithm presented for BMPs. However, if the radio ranges are not equal one or more edges may not be present dependent on the minimum radio range of the pair of nodes on that edge.

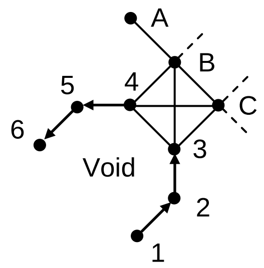

Figure 29: Crossed edges on polygon boundary.

To help illustrate the problems that may arise, and the complexity of determining what nodes to traverse, all possible configurations that may occur when any combinations of the four edges (other than the crossover) are not present are detailed in Figure 30.

Note that this is far from being a comprehensive list of configurations that may arise under these conditions, rather it is an example to illustrate the way in which the edge crossover problem becomes more complex in networks with non-uniform radio ranges.

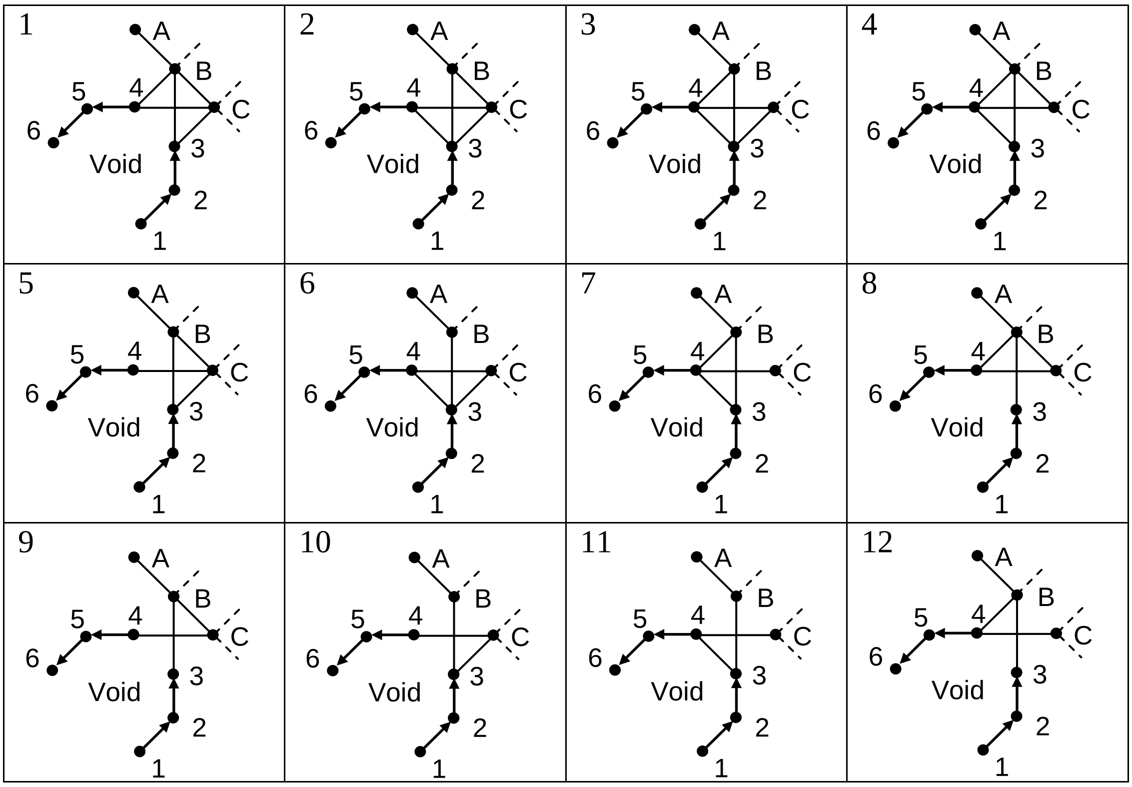

Figure 30: All permutations of crossed edges on void boundary.

Configurations 2, 3, 4, 6, 7, and 11 can be handled by the simple maze traversal algorithm outlined previously.

In the configuration 1, 8, and 12 the probe will traverse nodes 1, 2, 3, B, 4 but then will cross over the edge 3, B to node C and away from the polygon boundary. This is the basic edge crossover problem outlined previously for edge crossovers in a unit graph. The crossover can be detected using a simple directional area calculation on the potential edge 4, C and the vertices of the second last edge in the vector list 3 and B. The distance from the edge 4, C to each point 3 and B is signed to produce a signed area. If the sign of the area is the same then the point are on the same side of the edge. If the signs are different then the vertices lie on opposing sides of the edge and indicate a crossover. In the examples 1, 8, and 12 the nodes 3 and B lie on left sides of the edge 4, C and will be discarded and the next clockwise edge 4, 5 will be selected.

In the examples 5 and 10 a probe would traverse from 1, 2, 3, B and then incorrectly on to A if using the simple maze traversal algorithm. In example 9 a probe would traverse from 1, 2, 3, B and then incorrectly on to A.

4.4.2 Extended Local Knowledge

The illustrations above are over simplified in relation to the complexities that can occur with changes in topology and an increase in range differentials between high power nodes and low power nodes. For this reason probing will require additional information to make the correct next hop choices for boundary traversal.

The challenge is to extend the amount of local link state knowledge maintained by each node. The obvious solution is to maintain 2 hop neighbours and links. However this incurs an exponential increase in data size over the one hop neighbour information currently maintained.

To minimise this overhead it is proposed that only a single intermediate link is kept for each 2 hop neighbour. This will provide a limited view of the network and will require careful selection of the most appropriate neighbour and some heuristics to anticipate appropriate solutions to complex boundary configurations.

2 Hop Neigh Table Data Structure

4 bytes	NeighId
4 bytes	IntermediateNodeId
4 bytes	X coord,
4 bytes	Y coord
1 byte	isBidirectional
8 byte	Timestamp

With only a single intermediate node to each two hop neighbour the neighbour chosen must be the optimal incoming node into the 2 hop neighbour when considering the right hand rule for edge selection. The intermediate node is therefore selected as the node on the most clockwise (right hand) rotation about the incoming edge from the current node to the next hop neighbour as illustrated in Figure 31.

In Figure 31, F is a 2 hop neighbour of B. Of the candidate edges C-F, D-F and E-F the edge C-F provides the greatest clockwise rotation when entering F with the result that the next clockwise neighbour selected as the outgoing edge is correctly selected as F-G.

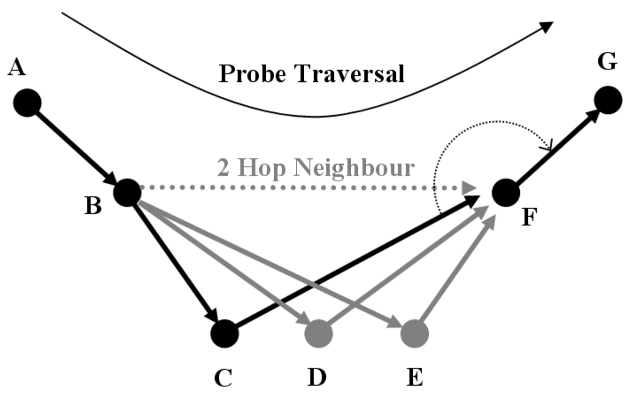

Figure 31: Two hop neighbour and intermediate node selection for neighbour table.

4.4.3 2 Hop Hello Message Packet Format

For routing protocols that require bidirectional links the hello protocol may include a list of neighbours (nodes from which it has received hello messages).

2 Hop Hello Packet Format

Node ID	Coord X	Coord Y	List Size	Neigh List
4 byte	4 bytes	4 bytes	1 byte	n x bytes

Figure 32: Extended 2 hop hello packet format.

Neigh List Entry for Hello Packet

4 bytes	Node ID
4 bytes	X Coord
4 bytes	Y Coord
1 byte	Is Bidirectional

For the two hop information required for BMP each hello message will also include the nodes location and a list of its directly connected neighbours, with extra fields for the location and the status of each link as shown in Figure 32.

4.4.4 Hello Message Processing Algorithm

This section presents the pseudo code used to process the 2 hop neighbour information and construct the neighbour table. In this algorithm the location and timestamp is updated for the neigh table entry if the neigh exists in the table, or a new entry if not. The new entry is initially set as unidirectional if not. Following this the neighbour's neighbour list is checked to determine if the neigh has received a hello message from the current node to indicate that the link is bidirectional.

Algorithm ProcessHelloMsg

Inputs: neigh, neigh location, neigh list /* list of neighs neighs (with node ID and loc) */
Outputs: None

/* 1. Process directly connected neighbour */

IF neigh exists in neigh table **THEN**
    update neigh location in table
    update neigh timestamp in table

ELSE IF neigh is not in neigh table **THEN**
    insert neigh in neigh table
    set neigh as directly connected
    set bidirectional as false
    set neigh location
    set neigh timestamp

/* now process the neighbour's neighbour list */

IF current node is in neigh's neigh list (it hears us, we hear it - so is bidir link) **THEN**
    IF neigh was a 2 hop neigh **OR** neigh link was **NOT** bidirectional **THEN**
        set neigh as directly connected
        set bidirectional as true

IF neigh is **NOT** directly connected **OR** neigh is **NOT** bidirectional **THEN**
    RETURN

/* 2. process 2 hop neighours */

/* check neigh's neigh list to check if it is an intermediate neigh to new 2 hop neighs */

FOR each node in neigh's neigh list **DO**

    IF link from neigh to neigh's neigh is bidirectional **THEN**

        /* add new bidirectional 2 hop neigh */

        IF node is not in this nodes neigh table **THEN**
            add node to this nodes neigh table /* node is neighs neigh */
            set location
            set timestamp
            set neigh as intermediate node

```
/* or replace a 1 hop unidirectional neigh with a 2 hop bidirectional neigh */
ELSE IF node exists in neigh table as directly connected unidirectional THEN
    set bidirectional as true
    set location
    set timestamp

    CONTINUE
/* or check if this is a better intermediate or just needs updating */
ELSE IF node exists in neigh table as a 2 hop neigh THEN
    IF via the same intermediate neigh THEN
        set location
        set timestamp

    /* check if new intermediate is better than existing intermediate */
    ELSE IF angle from 2 hop neigh to new intermediate
            > angle from 2 hop neigh to existing intermediate
        set neigh as new intermediate node
        set location
        set timestamp

    ELSE IF angle from 2 hop neigh to new intermediate
            = angle from 2 hop neigh to existing intermediate
        IF new intermediate is closer to 2 hop neigh than existing THEN
            set neigh as new intermediate node
            set location
            set timestamp
```

4.4.5 Modified Probe Forwarding Strategy

1. Deciding between one or two hop neighbour choice
2. Limiting decisions at an intermediate hop to prevent looping
3. Returning home
4. Partitioned networks

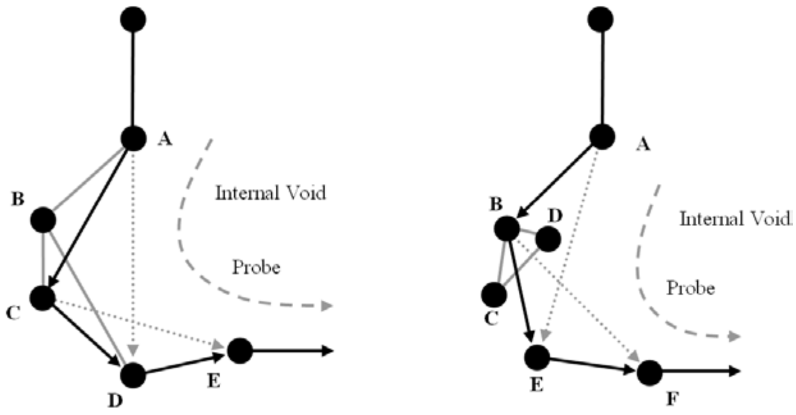

Figure 33: Selecting appropriate 2 hop probe traversal for (a) direct and (b) indirect links.

At the origin or at the destination of a single or two hop transition, either a single or two hop target node may be selected as that which is on the closer right hand rotation as shown in Figure 33.

At an intermediate node a probe can progress to a new two hop neigh of the current node or its two hop destination node, whichever is on the closer right hand rotation. It can not select a new single hop target from an intermediate node (even if it has the closest rotation) otherwise looping may occur as would happen in Figure 33(b) at node B.

Figure 34 shows the limited knowledge available for probe forwarding decisions at nodes A, B and E in Figure 33(b) when only one path to each two hop neigh is maintained.

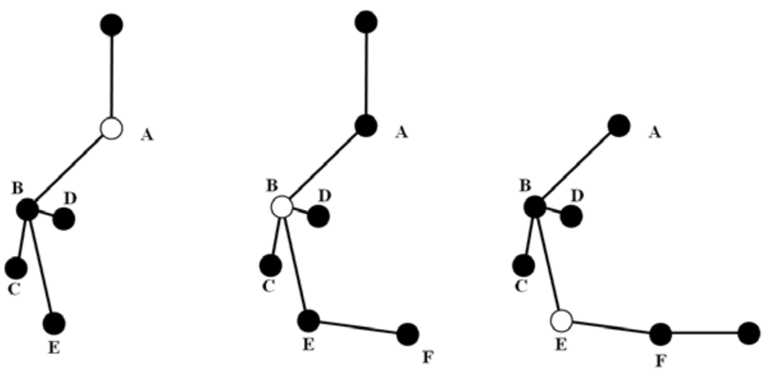

Figure 34: Local topology knowledge at nodes A, B and E.

At each node the incoming edge to be used as the reference from which the rotation is based is considered to be either the previous node or two hops previous dependent on which has the greater clockwise rotation.

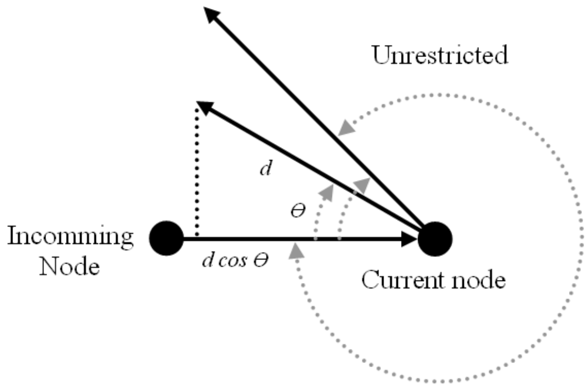

Figure 35: Limiting backwards probe packet traversal.

To address the problem of looping a restriction is placed on the next hop candidates at a rotation of less than 45° degrees from the incoming edge as illustrated in Figure 35. From 0° to 45° degree nodes next hop candidates are permitted if cosΘ of the distance to the next hop candidate is less than the distance to the incoming node.

The pseudo code for the *BackwardsCheck* algorithm is shown below

Algorithm BackwardsCheck

Inputs: target node, previous node
Outputs: result /* result = true if ok, false if potential for looping */

IF distance to target node < distance to previous node **THEN**
  RETURN TRUE

IF angle to target node > = angle to previous node + 45 degrees **THEN**
  RETURN TRUE

IF distance to target node * cos (angle to target node) < distance to previous node **THEN**
  RETURN TRUE

RETURN FALSE

4.4.6 Detecting Return to the Probe Source

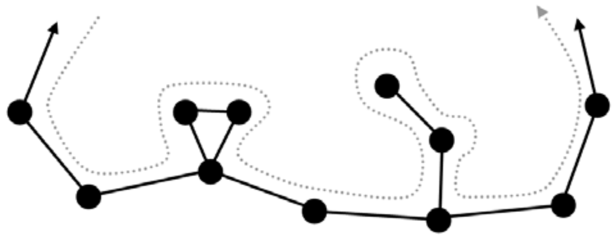

Figure 36: Nodes revisited on boundary traversal.

In boundary traversal a node may be visited multiple times if it is on a branch or at the junction of a segment as illustrated in Figure 36. If a boundary probe source node lies in a position that will be visited multiple times, then it can be difficult to determine whether a returned probe has completed the boundary traversal or is passing through, when its has only limited knowledge of the network topology.

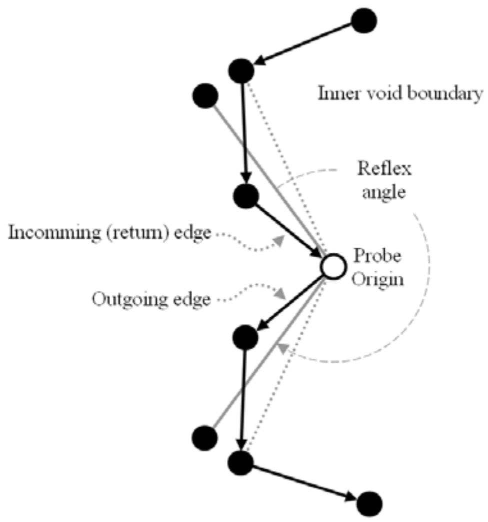

Figure 37: Probe initiation and return to source via 2 hop neighbours.

This is further compounded when either (or both) the first edge and the return edge, exit or return via links to intermediate nodes on two hop links as shown in Figure 37.

Another complication arises when the probe source is on a node on the ***outer boundary*** that segments the network. With limited knowledge of the network topology it is difficult to determine whether the probe has circumnavigated the network and has returned to the source via an indirect (two hop) link, or further network segment/s must be traversed, as shown in Figure 38.

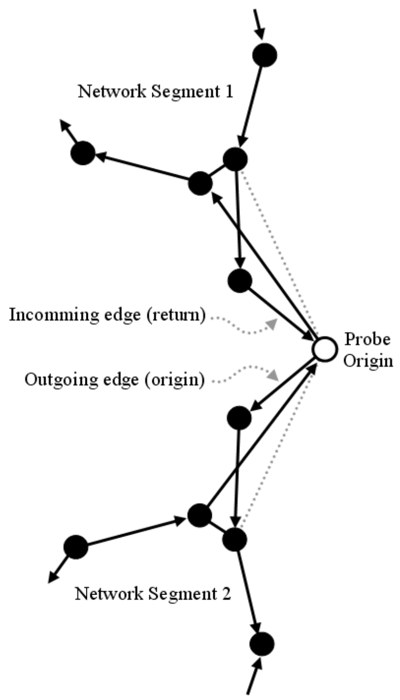

Figure 38: Probe source at a junction between network segments.

4.4.7 Back at Source Algorithm

There are two components to this problem.

1. Deciding whether the next hop could be a valid link home and restrict the next hop selection algorithm so that the source is not bypassed.
2. Deciding at the source node whether the probe has actually returned home or is passing through.

The following section details the algorithms proposed to deal with these problems.

The first algorithm *ReturningToSource* determines whether or not the next hop could potentially be a valid link which completes the boundary traversal, otherwise we may bypass it with the probe forwarding algorithm

Algorithm ReturningToSource

```
    Inputs: probe source, incoming node
    Outputs: forwardTo node, result          /* result = true if next hop is really home */

/* at an intermediate and next hop is home so just go there */

IF at intermediate node AND probe source = forwardTo node THEN
    forwardTo node ← 0
    RETURN TRUE

FOR each RH neigh (ordered by angle) in neigh table following on from incoming node DO

    /* don't consider home via an intermediate neigh or over a 2 hop link */

    IF link to RH neigh NOT bidirectional OR RH neigh is a 2 hop neigh THEN
        CONTINUE

    /* check if source is best (first valid directly connected) RH neigh */

    IF RH neigh = probe source THEN
        IF current node = minima LH node
            AND angle RH neigh from incoming node ≤ +/-45° THEN
            RETURN TRUE

        IF angle to RH neigh = angle to probe source node THEN
            RETURN FALSE

        RETURN TRUE

    /* if not at intermediate only check for source node up to the forward-to node */

    IF at intermediate node THEN
        IF RH neigh = forwardTo node THEN
            RETURN FALSE

    /* if not at intermediate only check for source node up to first valid RH neigh */

    ELSE IF NOT at intermediate node THEN
        IF current node NOT= minima LH node AND NOT HasXover(RH neigh)
            RETURN FALSE

RETURN FALSE        /* failed to find source */
```

The next algorithm is the *IsHome* algorithm which decides whether the probe has circumnavigated the void or outer boundary and is actually home (not passing through).

Algorithm IsHome

Inputs: next hop, previous node
Outputs: result /* result = true if probe has returned to source */

/* ok if not at nexus between network segments */

IF vertex left angle - vertex right angle < 45 **THEN**
    RETURN TRUE

/* make sure incoming edge is close to left vertice and outgoing is close to the right */

threshold ← vertex right angle + (vertex left angle - vertex right angle) / 2;
IF incomming angle >= threshold **AND** outgoing angle >= vertex right angle **THEN**
    RETURN TRUE

IF next hop = previous node **THEN**
    RETURN TRUE

RETURN FALSE

4.4.8 Get Next Hop Algorithm

The following algorithm will determine the next hop for probe forwarding. It will first determine which incoming node to use as a reference, check if the next hop will return the probe to the source, and exit if true.

It will then loop through the neighbour list (ordered by angle) starting from the incoming node and check each neighbour in turn until a valid edge is found. It first checks if the probe is at an intermediate node in traversal to a two hop neighbour. If so a new RH neigh can only be a 2 hop neigh (not a directly connected neighbour) otherwise looping may occur. If no better two hop neighbour is found the probe is forwarded on the final leg to the original two hop neighbour.

If not at an intermediate node and the next hop is a directly connected neighbour it is checked to ensure that it does not cross a previously traversed edge.

If the next best right hand neighbour is a two hop neighbour and we are at the intermediate node of a two hop traversal, it checks whether the new two hop neigh is not backwards (is within +/- 90) of the second leg of our current traversal. We also ensure that the intermediate node and two hop neighbour are not excessively backwards to the previous node to ensure looping does not occur.

If no neighbours are found and this is an intermediate node for a two hop traversal then continue on as planned.

The following is the *simplified* pseudo code for the *GetNextHop* probe forwarding algorithm

Algorithm GetNextHop

```
Inputs: incoming node, forwardTo node, probe source, probe path list
Outputs: next hop, forwardTo node

/* determine which node (last or second last) is the incoming node (has greater rotation) */

incoming node ← prev node        /* last or second last nodes extracted from probe path list */

IF prev Node NOT probe source node THEN
    IF secondLast node was NOT current node THEN     /* turn around on branch */
        IF angle to secondLast node > angle to prev node THEN
            incoming node ← secondLast node

/* check if probe has returned to source (not just passing through) */

IF ReturningToSource () THEN
    forwardTo node ← 0
    RETURN probe source as next hop

/* loop through right hand neighbours (ordered by angle) */

FOR each RH neigh (ordered by angle) in neigh table following on from incoming node DO
    IF link to RH neigh NOT bidirectional THEN
        CONTINUE

    IF RH neigh NOT incoming node AND incoming angle = RH neigh angle
        CONTINUE

    /* Condition 1 - at intermediate a new RH neigh can only be a 2 hop neigh (or continue) */

    IF at intermediate node THEN
        IF next best RH neigh is directly connected THEN
            IF neigh = forwardTo node THEN
                next hop ← forwardTo node
                forwardTo node ← 0
                RETURN next hop             /* just carry on as planned */

    /* Condition 2 - next best RH neigh is directly connected */

    ELSE IF RH neigh is directly connected THEN
        IF link to neigh does not crossover a previous link THEN
            forwardTo node ← 0
            next hop ← RH neigh
            RETURN next hop
```

```
/* Condition 3 - next best RH neigh is a 2 hop neigh */

        ELSE IF at intermediate node THEN

            /* skip if new 2 hop neigh is backwards from current node to prev 2 hop neigh */

            IF angle to RH neigh > angle to forwardTo node + 90
                OR angle to RH neigh < angle to forwardTo node - 90 THEN
                    CONTINUE

            /* skip RH neigh if backwards progression is excessive */

            IF MovesBackward (RH neigh) OR MovesBackward (intermediate node) THEN
                CONTINUE

            IF HasXover (RH Neigh) OR HasXover (intermediate node) THEN
                CONTINUE

            /* 2 hop RH neigh is useable */

            forwardTo node ← RH neigh
            RETURN intermediate node as next hop

    END LOOP

    /* failed to find a valid RH neigh */

    IF at intermediate node THEN     /* continue on as planned */
        next hop ← forwardTo node
        forwardTo node ← 0
        RETURN next hop

    RETURN FAIL                      /* probe forwarding process failed */
```

4.5 Boundary Confirmation - Storing Link State for Routing

When the probe packet arrives back at the source a boundary ***Confirmation*** packet will be sent back around the accumulated path list to confirm to each node that it is a member of the identified boundary. At each node the boundary node list is copied into the boundary record. The boundary information is then made available for routing.

An alternate solution was to store only left and right boundary neighbours and then use virtual coordinates to project the boundary node location onto a circular shaped virtual boundary. This would allow the improved forwarding strategy to successfully progress through local minima and traverse the boundary. Although this minimises memory overhead the problem there was no global knowledge to make informed decisions regarding the most effective direction of boundary traversal.

4.6 Protocol Implementation

This Section provides protocol implementation details for the proposed boundary mapping protocol BMP.

4.6.1 Boundary Probe Packet Format

Mode	Bdry ID	Bdry Src ID	Total Angle	List Size	Node ID List
1 byte	4 bytes	4 bytes	2 bytes	1 byte	$n \times 4$ bytes

Figure 39: Discovery and Confirmation mode packet header.

4.6.2 Boundary Record Data Structure

A node will keep a boundary record for each boundary for which it is a member. The source node ID will identify the root node that successfully originated and completed probing of the boundary. The boundary ID is the next sequential ID number assigned by the root node to each discovery probe that it initiates. These two numbers together uniquely identify any boundary in the network. The previous node is the node from which the probe for this boundary was received.

The timestamp field is the creation time, or update time if a lower probe ID is received and the boundary information is updated, or the time the last maintenance probe was received. The status field identifies the phase which may be *Discovery*, *Confirmation*, or *Stale*.

Boundary Record Data Structure

4 byte	Source node ID
4 byte	Boundary ID
4 byte	Previous node ID
4 byte	Next node ID
4 byte × n	Node ID list
8 byte	Timestamp
1 byte	Status

Status values may be *Empty*, *Discovery*, *Inner Boundary*, and *Outer Boundary*.

4.7 Results

Testing was performed on the same randomly generated network topologies as the previous tests. For evaluation of the boundaries, the number of boundaries, number of boundary hops, and the total probe byte count were measured. For memory and bandwidth, the hello byte count and the boundary record memory size were measured.

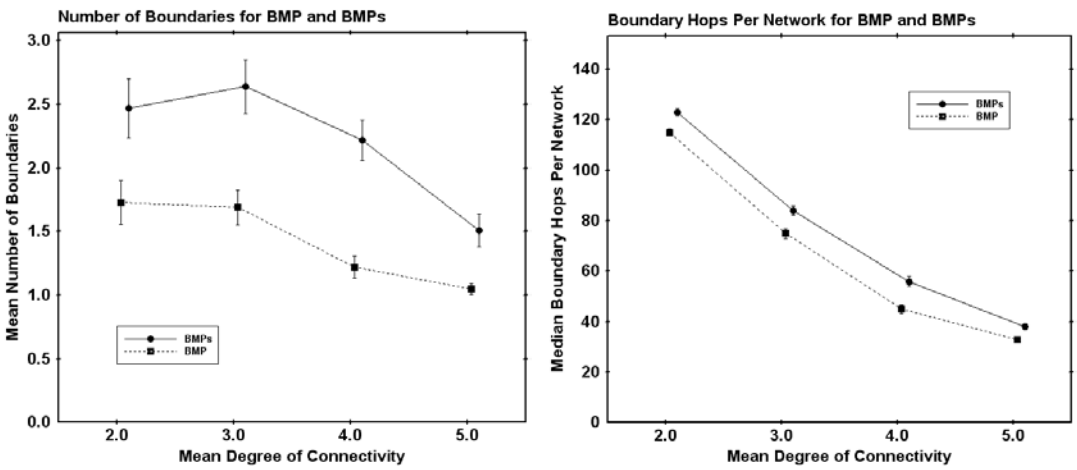

Figure 40: Mean number of (a) boundaries and (b) boundary hops for probing strategies.

The number of boundaries for BMP was less than BMPs as was the number of boundary hops (single circumnavigation) as shown in Figure 40(a) and (b) respectively. This to be expected as BMP did not probe boundaries with potential local minima that could not be dealt with using 2 hop neigh information.

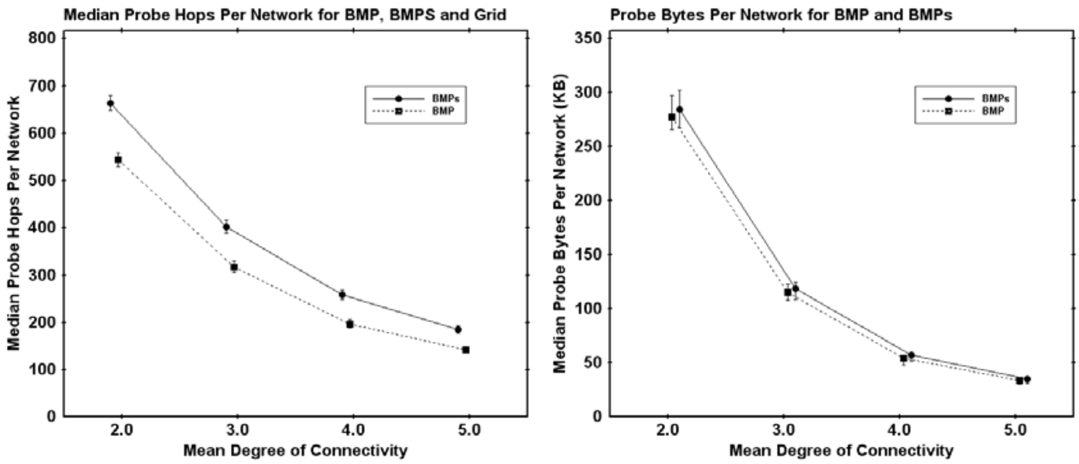

Figure 41: Median of (a) total probe hops and (b) probe bytes for BMP and BMPs.

Figure 41(a) shows the total number of probe hops per network. This included discovery and confirmation probes, in addition to the redundant boundary probes that were killed when duplicate probes were initiated on a single boundary.

The measure of interest is the overhead of probing in relation to the number of probe bytes required to successfully probe all boundaries. Even though BMP did not need to probe as many boundaries as BMPs this did not impact on the number of probe bytes as shown in Figure 41(b). None of the probe byte counts for BMP and BMPs was found to be significantly different using a using a Wilcoxon Signed Ranks test ($p=0.4862$, $p=0.3979$, $p=0.2168$, and $p=0.2374$ for a mean connectivity of 2.0 to 5.0 respectively).

The overhead for probing is quite varied (33KB to 285KB for BMPs and 33KB to 282KB) and reflects the total size of the boundaries in Figure 40(b).

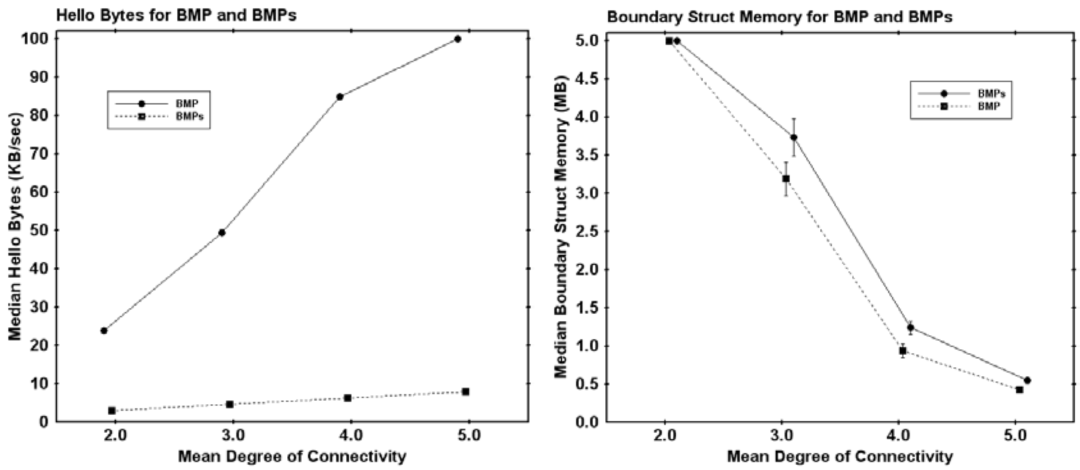

Figure 42: Median (a) hello packet overhead and (b) boundary struct memory overhead for BMP and BMPs.

The detailed two hop neighbour information required for BMP makes the hello packet overhead significantly higher for BMPs as shown in Figure 42(a) (note that confidence intervals are extremely small). This is the cost of including the additional absolute node location for all neighbours in BMP whereas BMPs only includes node ID for identification of bidirectional links.

The memory cost for both protocols is also extremely high with a maximum boundary data size of 4.9 Mb in the sparse networks which had longer boundaries as shown in Figure 42(b). This is a serious problem that will be addressed in Chapter 6. However it is important to continue on and apply boundary mapping (as developed in this chapter) for use in a geographic routing protocol to test the effect of making informed decisions for direction of boundary traversal at local minima in comparison to a fully distributed geographic routing protocol. This is essential in determining the effect of the availability of limited global information on routing protocol performance.

4.7.1 Boundary Probing with Variable Radio Ranges

For this test 100 network topologies were generated with radio ranges that vary in the ratio of 1:√2 as proposed in [49]. The lower limits of radio range were 125 meters, 150 meters, 175 meters and 200 meters with 100 nodes randomly placed in a 1km x 1km area.

Table IX: Percentage of Networks with Probing Failures

	Networks with Probing Failures			
Strategy	125 metres+ (%)	150 metres+ (%)	175 metres+ (%)	200 metres+ (%)
BMPs	11	8	7	0
BMP	0	0	0	0

Table IX shows the number of networks where probing failed under a condition of variable radio range of 1:√2. BMP which was designed to accommodate some complexity due to complex edge crossovers did not exhibit any probing failures. In contrast BMPs as expected failed to successfully probe all boundaries.

4.8 Discussion

Although the use of boundary state information was shown to be effective for routing around local minima, the cost of maintaining this information was very high, with boundary data sizes up to 4.9 Mb in the test networks. This a serious problem in mobile networks where devices are typically lacking in resources. In addition to this problem, the improved BMP algorithm will only deal with limited variability in radio range and its heuristic approach can not provide any guaranteed level of performance.

To address the excessive control and memory overhead, a low resolution occupancy grid based approach to boundary mapping is investigated in Chapter 6 as an extension to this study. As this problem is related to the efficiency of the mapping protocol it will not effect the application and development of boundary state routing algorithms which will use boundary state information for routing around local minima.

In the next chapter the Boundary State Routing protocol (BSR) will be presented which uses the boundary state information maintained by BMP.

Chapter 5

Boundary State Routing

It is remarkable that Barrallier should have followed so far down the Kowmung before turning to the left, for had he turned up the river instead of down he would probably have succeeded in crossing the Great Dividing Range, after which he would have had no difficulty in proceeding westward.

Ida Lee
Early Explorers in Australia, Ensign Barrallier, 2002

This chapter investigates the use of boundary state information for routing around local minima. This involves choosing a direction of boundary traversal for the optimal exit point and loop prevention when switching between multiple strategies.

5.1 Path Selection for Boundary Traversal

The advantage of using boundary link state information for routing is that an informed choice can be made regarding the best direction of boundary traversal. The development of an algorithm required for path selection presented a range of issues.

- Choosing a boundary exit point
- Choosing the best (shortest) path (and direction) to the boundary exit point
    - Pruning branches from the path
- Swapping between boundaries
- Ensuring looping does not occur
- Switching between strategies (boundary state and the primary forwarding strategy or strategies) without looping

5.2 Extension to BMP - Boundary Exit Point and Next Hop

When in boundary node the optimal boundary exit point for the route destination will be computed as the node closest to the destination. In some instances this will not be the node with the shortest path to the destination, however without further information beyond the current boundary there is no other option than choosing the closest node. When the data packet is in transit around the boundary, the exit point will be re-evaluated at each boundary node and the packet may also transfer to another boundary (where a node is a member of multiple boundaries) if that boundary has an exit point closer to the destination than the current exit point.

The data packet will exit the boundary when a node is available that is closer to the destination than the closest ever location, in which case the packet will revert to Greedy forwarding.

After selection of the optimal exit point, the optimal direction of traversal is determined based on the shortest path around the boundary (with branches pruned).

Note that the exit point and next hop are calculated within BMP and the next hop is made available to BSR.

5.3 Boundary State Routing

The proposed routing protocol called Boundary State Routing (BSR) is implemented using a combination of Greedy-BoundedCompass forwarding with routing at local minima performed using boundary state information maintained by the boundary mapping protocol detailed in the previous section.

5.3.1 Loop Prevention in Multi-Mode Routing Strategies

In the proposed multi-strategy routing protocol, it was found that looping may occur when changing between strategies if the packet has moved away from the destination. This occurred because routing decisions can not take into account the paths traversed by previous strategies (without accumulating path information in the packet header). The problem was solved using the progressively reset at closest ever location used by Greedy-BoundedCompass and applying the restrictions below.

Rules applied to the routing decision process:

1. Each individual forwarding strategy must not loop.
2. The primary forwarding strategy must always move closer to the destination
3. After switching from the primary forwarding strategy, a packet can only move forward through an ordered set of available strategies
4. A packet can only switch back to the primary forwarding strategy for the next hop node that is closer to the destination than the closest ever location, at which stage the defined sequence of strategies may begin again and the packet may again move away from the destination.
5. If the final strategy in the sequence fails and no next hop node exists that is closer than the closest ever location then the packet is dropped.

5.3.2 Boundary State Routing Algorithm

The boundary state routing algorithm is presented below. When a packet is to be routed from the source or an intermediate node, BSR will first attempt to route the packet using Greedy forwarding, regardless of the current routing mode setting in the packet.

If Greedy forwarding fails and the packet is not in Boundary mode, BSR will check for a route using BoundedCompass forwarding. If successful and the next hop is closer to the destination than the current node then the BoundedCompass route is used. If the next hop is further from the destination, the algorithm checks for an alternate boundary route (i.e. the current node is on a boundary containing a node closer to the destination than the current node). If successful the boundary route is used in preference to the BoundedCompass route as the choice is informed by the optimal direction around the boundary. If unsuccessful the BoundedCompass route is used.

If instead, both Greedy and BoundedCompass forwarding fail, the algorithm will check for a boundary route. If it fails to determine a suitable next hop then the packet will be dropped, otherwise the packet will be forwarded to the next hop.

5.3.3 Boundary State Routing Pseudo Code

The *simplified* BSR routing algorithm pseudo code for the *BSRRouteQuery* algorithm is detailed below.

Algorithm BSRRouteQuery

Inputs: destination node
Outputs: next hop

IF destination is directly connected neighbour **THEN**
    RETURN destination as next hop

next hop ← **Greedy()**

IF next hop = FAIL **AND** mode **NOT** = BOUNDARY **THEN**
    next hop ← **BoundedCompass()**

IF next hop = FAIL **AND** mode = COMPASS **THEN**
    IF distance from next hop to destination > current node to destination **THEN**
        alternate next hop ← **Boundary()**
        IF alternate next hop **NOT** = FAIL **THEN**
            next hop ← alternate next hop

IF next hop = FAIL **THEN**
    next hop ← **Boundary()**

RETURN next hop

The following is the *simplified* pseudo code for the *Boundary* mode algorithm called from the *BSRRouteQuery* algorithm.

```
Algorithm Boundary
    Inputs: destination node
    Outputs: next hop

    FOR each boundary DO
        IF boundary mode NOT = DISCOVERY THEN
            FOR each node in boundary node list DO
                IF boundaryNode is closer to dest coord than exit node THEN
                    exit node ← boundary node
                    boundaryId ← boundary id

    candidateNode = getLeftRoute(boundaryId, exit node)      /* shortest path in pruned list */

    IF hop count to candidate node < hop count THEN
        hop count ← hop count to candidate node
        next hop ← candidate node

    candidate node = GetRightRoute(boundaryId, exit node)    /* shortest path in pruned list */

    IF hop count = 0 OR hop count to candidate node < hop count THEN
        hop count ← hop count to candidate node
        next hop ← candidate node

    RETURN next hop
```

5.4 Protocol Implementation

This section provides details of the protocol implementation and test results for the proposed boundary state routing protocol BSR.

5.4.1 Data Packet Headers

There are two data packet types which are used for Greedy mode, Compass mode, and Boundary mode. The packet type for each strategy is identified by the value in the mode field in each packet.

Mode	Prev ID	Closest X	Closest Y
1 byte	4 bytes	4 bytes	4 bytes

Figure 43: Greedy and Boundary mode packet header.

Mode	Prev ID	Closest X	Closest Y	Angle
1 byte	4 bytes	4 bytes	4 bytes	2 bytes

Figure 44: Compass mode packet header.

For the Greedy and Boundary mode packet header in Figure 43, the closest location (X, Y) is always the previous node location for Greedy as a packet can only move closer to the destination. For Boundary mode it is the closest location (X, Y) which is the coordinate of the closest node to the destination along the path that the packet has traversed.

For both Compass mode and Boundary mode the data packet can not revert to Greedy mode unless a next hop node exists that is closer than this location.

The Compass mode packet header in Figure 44 includes an additional field for the cumulative angle traversed. This field is a 16 bit unsigned integer which holds a scaled fractional value for the angle traversed which uses a scale factor of 100. In Compass mode a next hop candidate will be excluded if traversal to that node would exceed a traversed angle of +/-90°. Although this rule should exclude the previous node, the previous node ID is included in case adjacent nodes in the path contain inaccurate location information.

Note that packets may also include the destination location. This would be depend on the type of location management scheme used. Note that for simulation tests in the following section, the presence of a destination location database is assumed.

5.5 Results

Testing was performed with the same methodology used in section 3.8 for path completion and path efficiency (measured as the ratio of the shortest path hop count to the actual hop count for each path). More detailed results are provided for the path efficiency and maximum hops at individual path lengths to investigate where the difference in performance occurs.

To control for forwarding strategy when comparing the performance of BSR to GPSR, BSR was also compared with the improved GPSR with Greedy-BoundedCompass forwarding to evaluate the comparative improvement in path efficiency from the use of boundary state information for routing decisions.

5.5.1 Path Efficiency Comparison All Paths

Results for path completion indicate that both protocols achieved 100% path completion confirming the absence of looping.

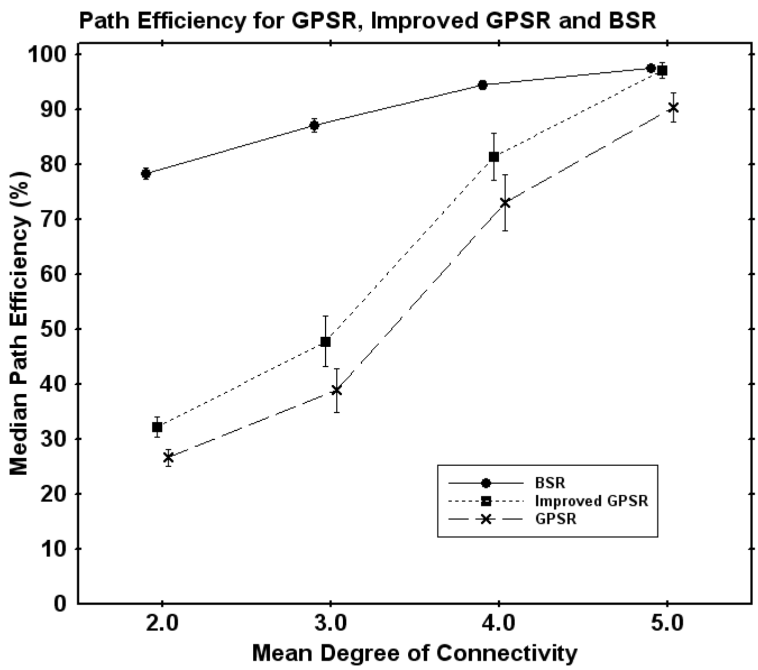

Figure 45: Path efficiency for BSR, GPSR, and improved GPSR.

It is clear from the values and confidence intervals in Figure 45 that BSR has a significantly higher path efficiency than both standard GPSR and improved GPSR. The improvement was 46.1%, 39.4%, 8.5%, and 0.5% in the median of paths completed at a connectivity of 2.0, 3.0, 4.0 and 5.0 respectively (all significant at $p < .01$ using a Wilcoxon Signed Ranks test).

5.5.2 Path Efficiency Comparison for Individual Path lengths

The next section discusses in more detail the effect of informed decisions regarding boundary traversal on path efficiency.

Figure 46 to Figure 49 shows the median hop count at a connectivity of 2.0, 3.0, 4.0 and 5.0 for GPSR, GPSR with Greedy-BoundedCompass, and BSR, for all shortest path lengths which ranged from 1 up to 35 hops in the sparse networks.

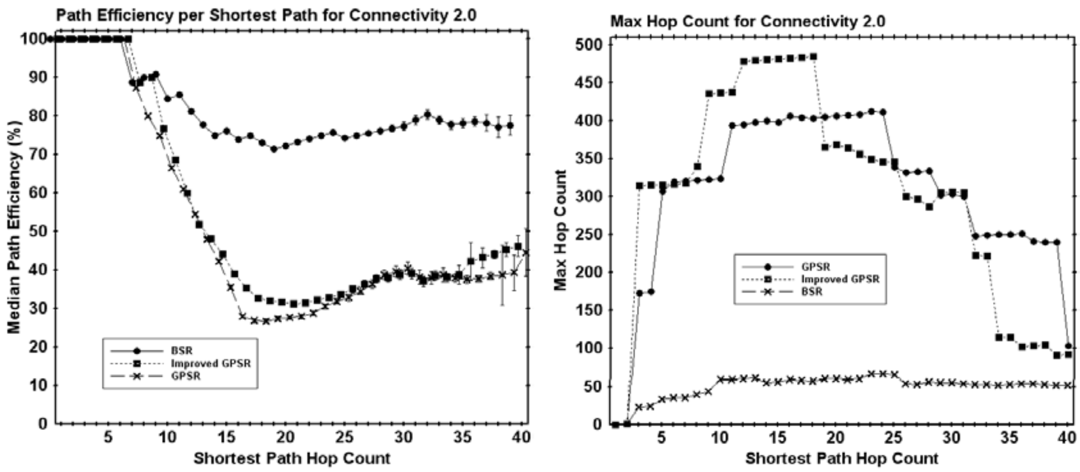

Figure 46: (a) Median path efficiency and (b) max hop count for each shortest path length for a mean connectivity of 2.0.

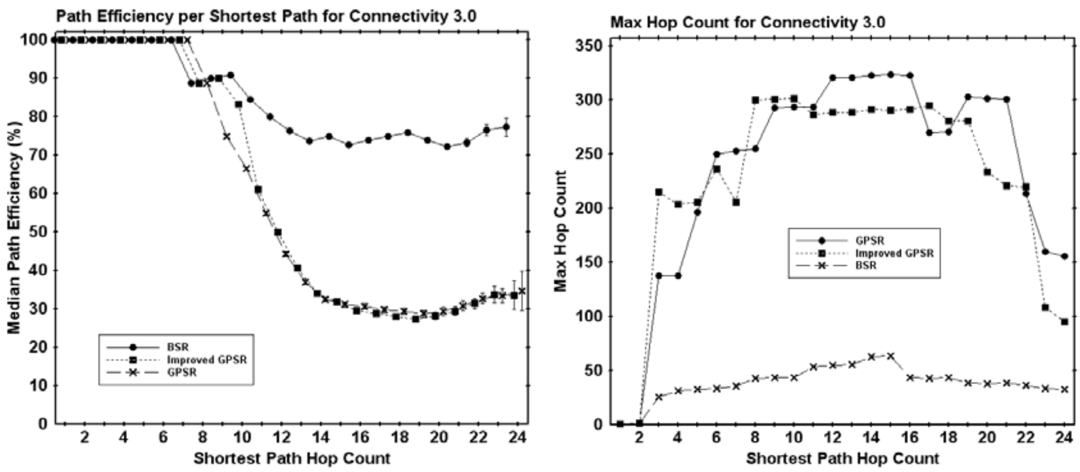

Figure 47: (a) Median path efficiency and (b) max hop count for each shortest path length for a mean connectivity of 3.0.

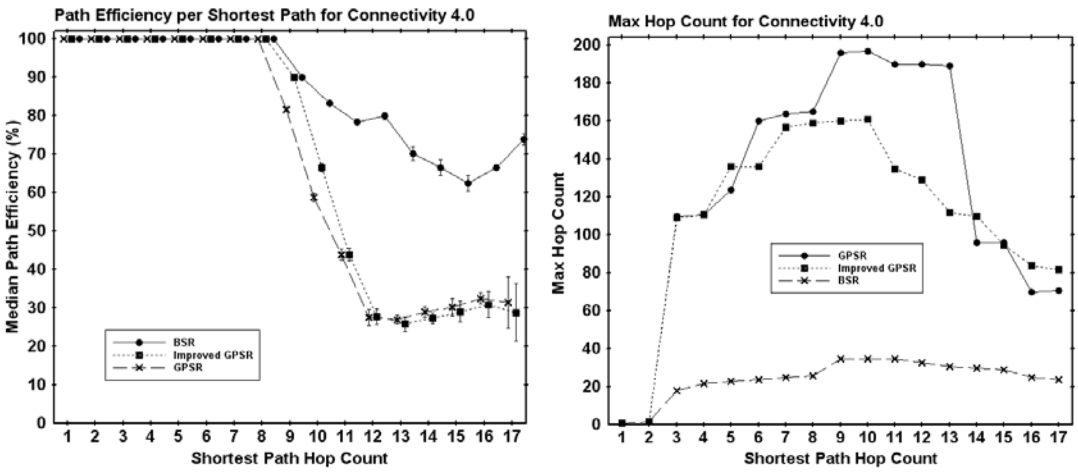

Figure 48: (a) Median path efficiency and (b) max hop count for each shortest path length for a mean connectivity of 4.0.

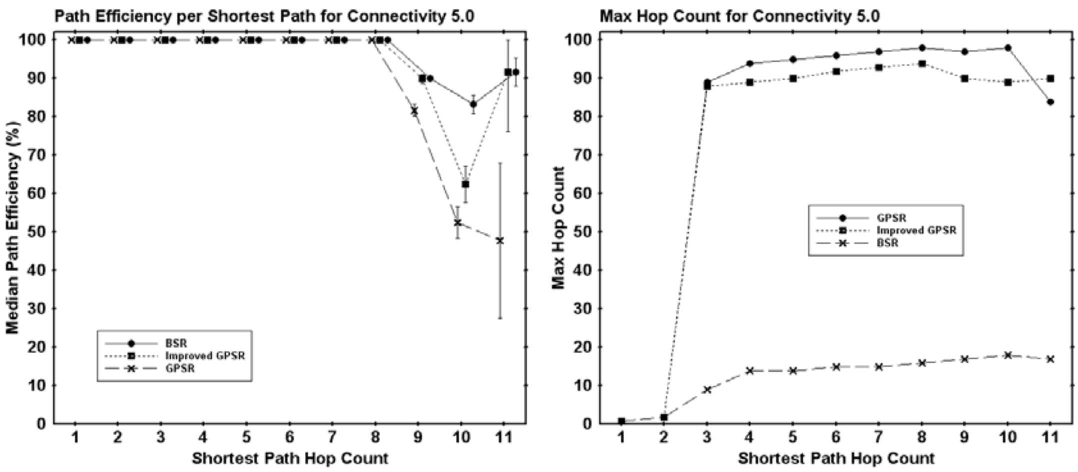

Figure 49: (a) Median path efficiency and (b) max hop count for each shortest path length for a mean connectivity of 5.0.

At very low path lengths routing choices are made using link state entries for directly connected neighbours or by Greedy forwarding, and so all protocols performed close to the shortest path length as shown by the median path efficiency. As the path length increases the reason for the divergence in the hop count is explained by the extremely indirect routes taken by the two versions of GPSR due to uninformed decisions at local minima regarding direction of boundary traversal of the planar sub graph in perimeter mode. An example of these multimodal distributions are shown in more detail in Figure 50 and help explain the reason for the strange results for improved GPSR at a shortest path hop count of 11 for a connectivity of 5.0.

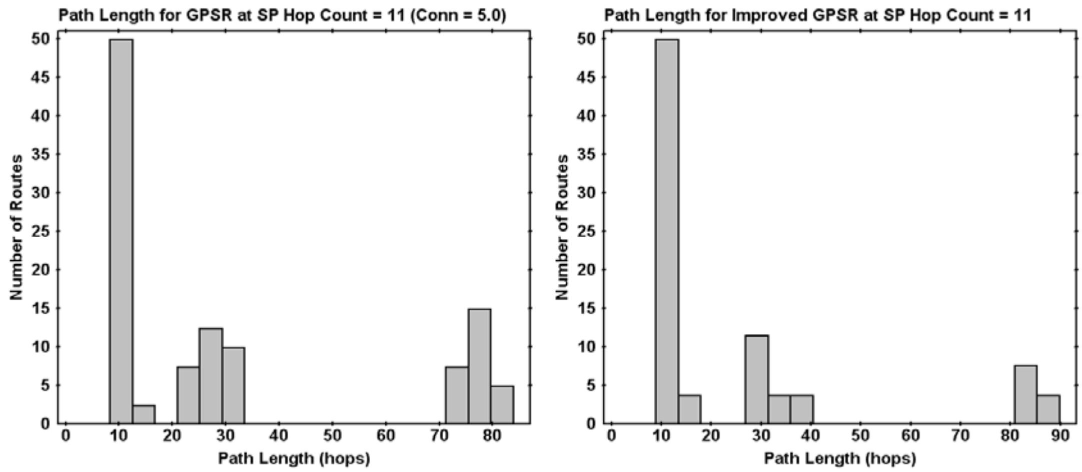

Figure 50: Distribution of route length for GPSR and improved GPSR at a shortest path length of 11 (at a mean connectivity of 5.0).

In contrast to GPSR which uses a distributed algorithm and does not maintain any global knowledge, BSR performs far more consistently across the range of shortest path lengths. Where GPSR has maximum path lengths in the order of 400 to 500 hops, BSR has maximum hop counts in the order of 50, showing the effect of using boundary state information for routing decisions.

5.6 Discussion

The capability of Greedy-BoundedCompass forwarding to handle boundary nodes with non-reflex angles reduced the number of network nodes that were required to initiate boundary probes to nodes having a reflex angle only. The boundary state routing protocol BSR which used Greedy-BoundedCompass and boundary state information from BMP confirmed the advantage of using limited global information.

However, the cost for BMP in terms of bandwidth and especially memory requirements for boundary data structures (max 4.9 MB) was shown to be excessive. In addition, BMP has a critical reliance on accurate node location and edge orientation. These are serious problems that have prompted further research into an alternate solution which is presented in the following chapter.

Chapter 6

Grid Occupancy Mapping

> *It also calls out to traditions in cultural studies and ethnography of self-reflexive research practices and an insistence upon the acute contextualisation (local, national, global) of our various sites of interest and our own investments in them.*
>
> *Katrina Schlunke*
> *Historicising Whiteness: Captain Cook Possesses Australia*
> *Historicising Whiteness Conference, Melbourne, 2006*

The use of global information has been shown to significantly improve routing decisions in relation to the performance of a distributed geographic routing protocol that relies on local information only to make boundary traversal decisions at local minima. However the overhead of probe packets, data structures (memory), and complex processing algorithms is a severe limitation of the proposed protocol. Also the reliance on accurate node location and link angle makes it difficult to migrate BMP to dynamic networks under conditions of mobility.

A more appropriate mapping solution is therefore required that provides global knowledge about critical areas of the topology, and balances the need for accuracy and detail, while minimising bandwidth (probe size), memory (number, size and amount of information stored in boundary data structures), in addition to minimising processing requirements.

6.1 Problem Definition

To remove the dependence on accurate node and link information it is proposed that probing and mapping of the network topology use a grid system with each cell of the grid representing occupancy. Probes can then be propagated along occupied boundary cells, thus minimising the need to maintain accurate node information relating to individual nodes. The granularity of the topology map can be optimised to minimise overhead while ensuring that the resolution is sufficient to adequately define the relevant features of the network topology that will impact on routing decisions.

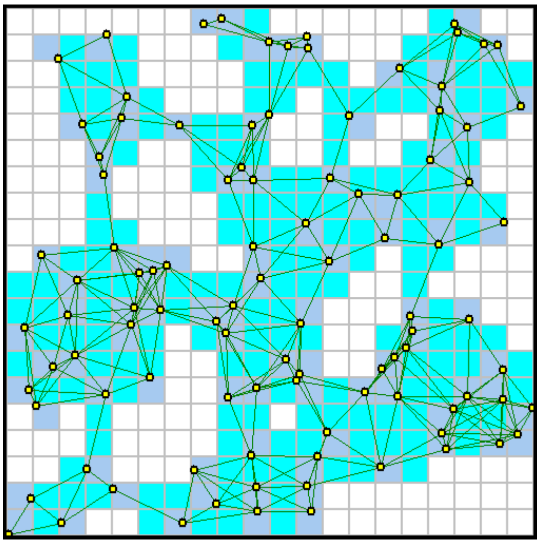

Figure 51: Global view of the grid occupancy map for a 20 x 20 grid.

From a global perspective the cell occupancy grid is shown in Figure 51 (note that this is global view that is not fully represented in individual nodes).

The occupancy grid is the only data structure in which information is maintained. Each cell within the occupancy grid will be represented as:

- isOccupied — By a directly connected neighbour
  Stores list of neighbours in cell with entries aged & deleted
- isLink — Coverage by a link to a directly connected neighbour
- isRemote — Neighbour coverage (of indeterminate type - node or link)
  Stores list of neighbours that provide access to this cell
- isBoundary — Is a member of one or more boundaries

6.2 Optimal Cell Size

Cells may be any minimum size but too small a resolution will mean excessive detail and information will be stored at each boundary cell and propagated on boundary discovery.

The maximum size is governed by the requirement that any two cells that can not communicate MUST be separate by an empty cell. Cell size needs to be small enough so that two nodes operating at the minimum radio range and at the furtherest distance apart where they can communicate will still be in adjacent cells of the grid if placed in the opposing corner of their cells.

If these cells are any further apart (no longer communicating) they must be separated by an empty cell (no longer in adjacent cells).

For the example below a minimum radio range of 150 meters will be used.

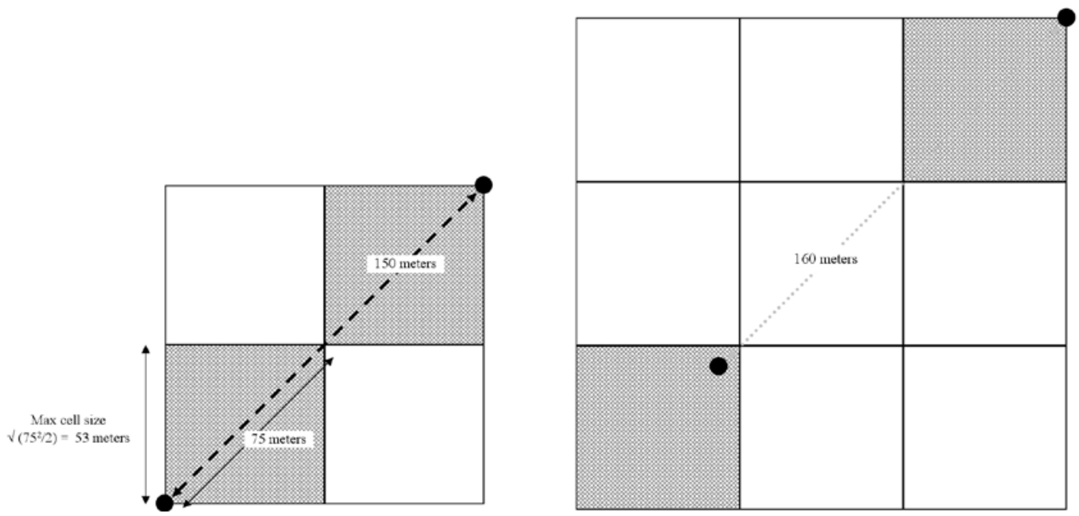

Figure 52: Maximum grid size calculation.

For a minimum 150 meters radio range, as illustrated in Figure 52, the maximum cell size is $\sqrt{(150/2)^2/2}$ = 53.0 meters. For practicality, this has been reduced to 50 meters for the examples to follow (the cell size can be smaller but can not be larger).

Note that as per previous testing a network of 1km x 1km is used. With a minimum radio transmission range of 150 meters and a cell size of 50 meters the grid dimensions will therefore be 20 by 20 (400 cells total).

6.3 Smoothing Effect of Lower Resolution

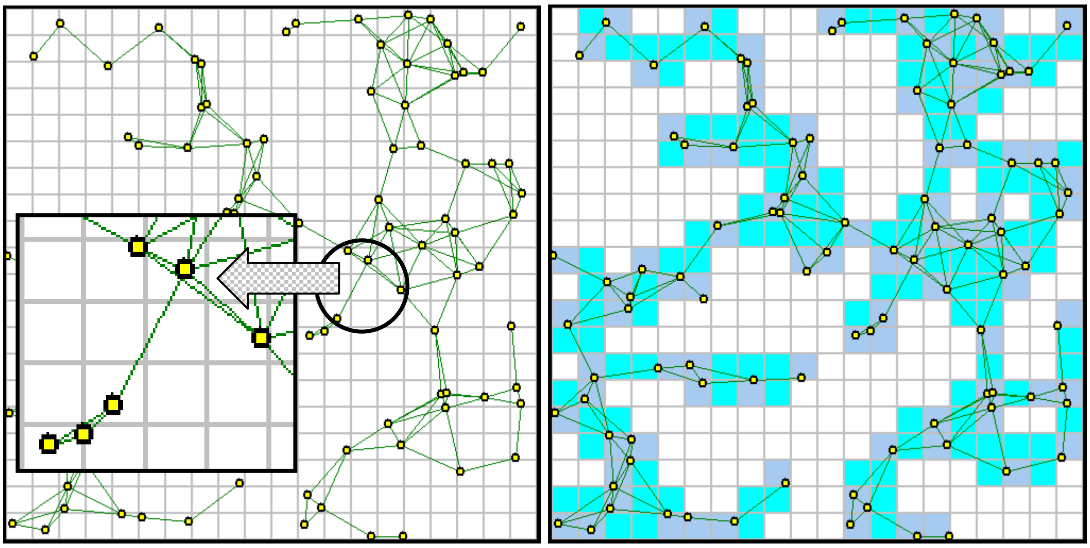

Figure 53: (a) Link crossover on boundary missed by BMPs and (b) the alternate low resolution occupancy grid representation.

The smoothing effect of the lower resolution can be seen in Figure 53(b) (a random network that arose during testing). In this network where the crossover link highlighted in Figure 53(a) is not directly visible by a node maintaining one hop neighbour information only, the adjacent boundary nodes will be traversed and the branch will be bypassed. This will result in the branch nodes becoming inaccessible to routing from some directions.

6.4 Local Cell Occupancy Exchange (Hello Information)

To minimise overhead, nodes will exchange their local (directly connected) cell occupancy by appending it to their hello messages. From the received information they will maintain, at a minimum, 2 hop local occupancy information in their full network grid occupancy map.

To minimise the extension to the hello packet, cell occupancy is simply a 1 or 0 depending on whether a cell is occupied or not (by a directly connected neighbour or a link to a directly connected neighbour).

For a cell size of 50 meters and radio range of 150 meter the maximum cell range will be (horizontal or vertical) 150/50 = 3 cells, plus 1 if at the left or right extreme of the cell, which is a maximum cell size of +/- 4 cells = 8 cells.

This can be conveniently (and efficiently) represented using 1 byte for an 8 cell row with each bit representing the cell occupancy. For 8 rows by 8 columns this will require 8 x 8 bytes = 64 bytes for the local occupancy grid map extension to each hello packet. This size will apply to any radio ranges as the optimal cell size is always the same ratio of cell size to minimum radio range.

6.4.1 Local Occupancy Advertisements

When constructing a local occupancy advertisement a node will only insert cell occupancy of directly connected nodes or edges which represent the trusted portion of their own local occupancy map. It will ***NOT*** include the occupancy of remote cells (more than one hop away) that have been learnt from a neighbouring node.

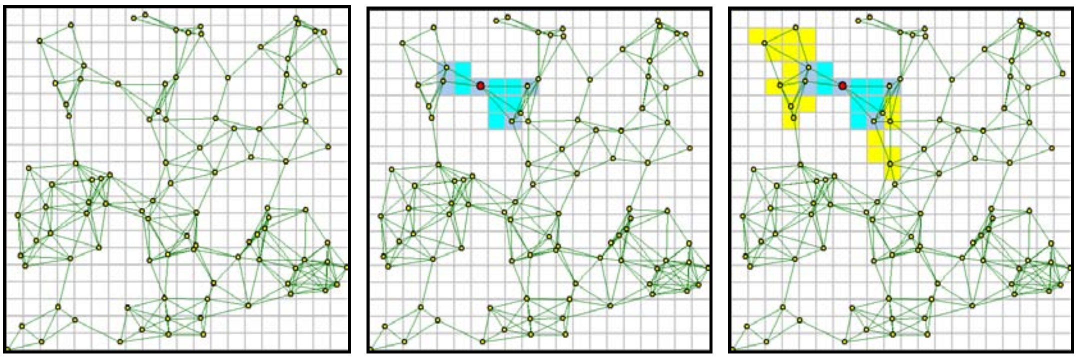

Figure 54: (a) Initial network, (b) trusted local occupancy, and (c) remote occupancy learnt from neighbours (for highlighted node).

It is easy to forget that each node in the network has only a limited view of the network, and will have little knowledge of the full range of nodes and links as illustrated for the network in Figure 54(a).

Figure 54(b) shows the local node occupancy and link coverage in the grid occupancy map for the selected cell. This represents directly connected neighbours in the trusted portion of the local occupancy map after one round of neigh exchanges to establish that links are bidirectional.

This trusted portion of the occupancy map is then appended to the hello messages which are broadcast to neighbouring nodes. Figure 54(c) shows the updated occupancy map after the node has exchanged extended hello messages with its neighbours. This additional remote information is ***not*** included in the local occupancy hello updates as it is not considered trustworthy. The remote (two hop) information is used to determine the specific cell and quadrant where trusted local information can be relied upon for boundary probe forwarding. It is not used to determine the boundary path as there is no way of determining whether adjacent cell occupancy information is missing and a cell is assumed to be empty when it is occupied.

6.5 Probing Using Trusted Local Occupancy Information

Boundary probes are initiated from any node with a reflex angle as in BMPs. The next boundary cell is selected as the cell on the right hand rotation from the local minimum (i.e. uses the right hand rule). The probe is forwarded to the first occupied cell with the consideration that the closer the next hop the higher the probability that an unforeseen change or missing information in the occupancy map can be corrected.

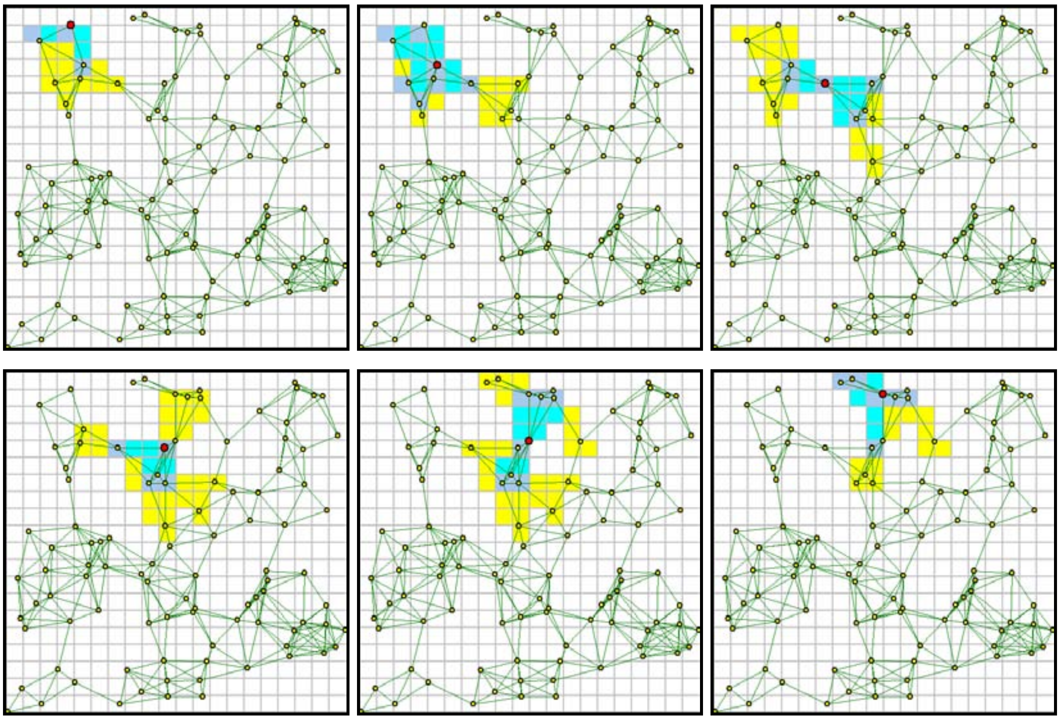

Figure 55: Progression of boundary probe – local view from each node.

Figure 55 illustrates the progress of a probe initiated in the top left hand corner, and shows the local and remote knowledge available for each progressive probe forwarding decision (which uses trusted local one hop knowledge only).

When the right hand rule reaches the limit of trusted knowledge and does not find an occupied cell, it stops at the first cell flagged as remotely occupied and sends the probe packet to the intermediate node who passed on the information that this cell was occupied in its hello message. When the probe is forwarded to the intermediate cell the remotely occupied cell is not appended to the cell path list. At the intermediate cell the boundary progress is re-evaluated from the last trusted cell.

6.5.1 Occupancy Grid Probe Packets

ProbeType	EntrySector	Position	NumCells	BoundaryCell List
1 byte	1 byte	1 byte	1 byte	$n \times 2$ bytes

Figure 56: Occupancy grid probe format.

Figure 56 shows the packet header format for the occupancy map probe packets.

Below is the description of the fields.

Probe type	Discovery or Confirmation
EntrySector	Incoming sector at local minima
Position	Position of current node in list
BoundaryCells	Cell entry format - row (1 byte), column (1 byte)

6.5.2 Probe Records

Below is the data structure for each probe record. As each probe passes through a boundary cell, a record is made of the probes visit using the cell offset (row * maxcols + col) as the probe ID.

Probe Record Data Structure

2 bytes	ProbeId	
1 byte	IncommingVector	8 segments (0-7)
1 byte	OutgoingVector	
1 byte	State	Source, Killed, Discovery, Confirmed

6.5.3 Detecting Home

The processing for the grid occupancy map is far simpler than for BMP, but some aspects are still relatively involved. In occupancy grid probing the probe home problem is addressed using a sliding window as illustrated in Figure 57. This system evaluates either the previous path or the predicted boundary path and uses a threshold to confirm a match to the starting segment of the path.

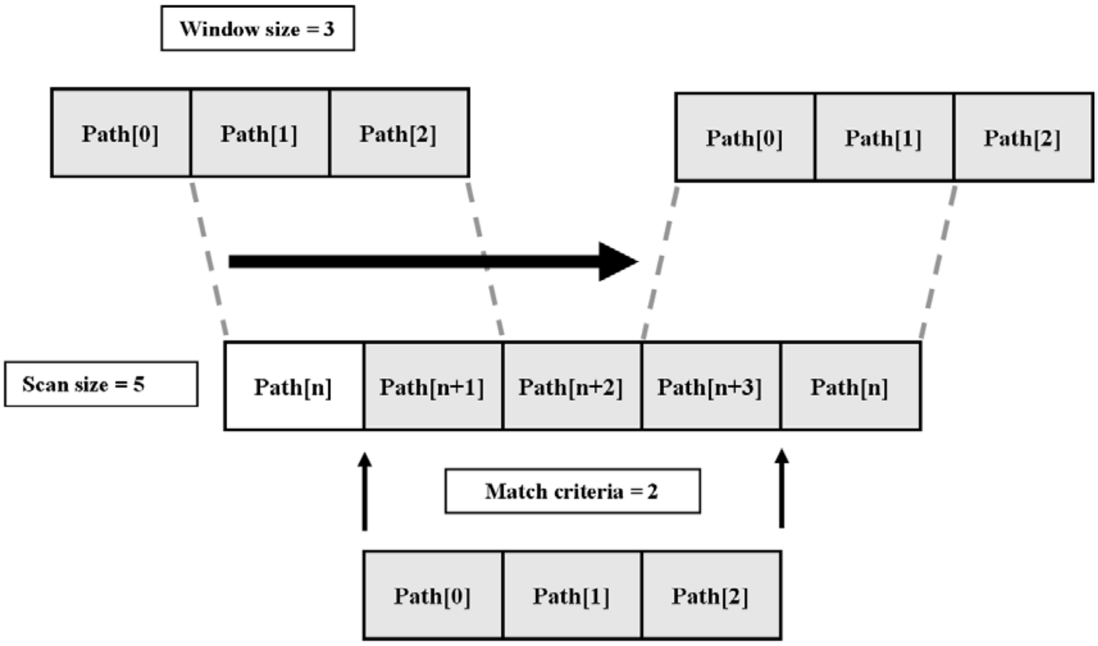

Figure 57: Sliding window for home detection.

This process was developed because grid occupancy does not deal with individual nodes and a return to the source cell may not mean returning to the originating node. This method has some advantages. If a probe packet does not stop within (may bypass) the source cell there is built in tolerance for overshoot. The nodes within the source cell will still be aware of the passing and update any related boundary cells and records appropriately. The node detecting the overshoot will detect the passing of home and change the packet mode to confirmation (or drop the packet if probing is complete).

This again is an example of how the lower resolution grid representation makes processing simpler than the node and edge approach of BMP.

6.6 Confirming/Distributing Boundary Information

Because detailed boundary path information is not distributed, only cell occupancy which updated the occupancy map, the confirmation packet can delete traversed cells as it circumnavigates the boundary in confirmation mode, which reduces the overhead of probing.

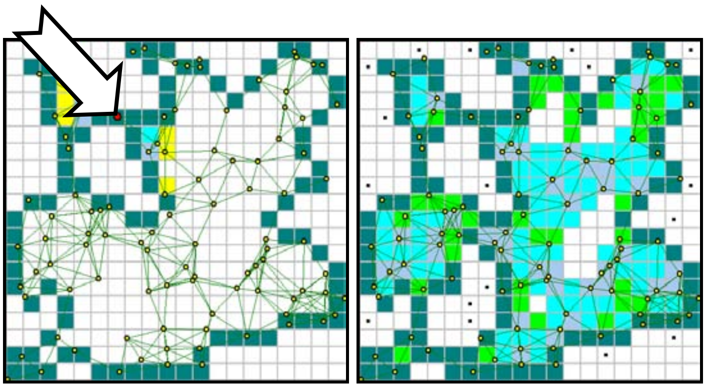

Figure 58: (a) Boundary map at highlighted node after probing is complete and (b) the global occupancy map.

After successful discovery the packet must traverse the path defined within the discovery packet to confirmation the boundary. The packet moves cell by cell along boundary path where the cells are occupied, and moves via intermediate nodes for empty cells (flagged as occupied because of coverage by a link between nodes).

Figure 58(a) shows the final view of the network from the indicated cell. Figure 58(b) shows a global view of the grid occupancy map (artificially constructed from the sum of local views).

6.7 Looping and Complexity

The processing requirements for the lower resolution approach of the occupancy grid are much simpler. This is further improved because this system is inherently free from looping so there is no need to scan back though the path for each candidate next hop to look for crossovers as with BMP and BMPs which greatly reduces the processing complexity.

6.8 Results

Testing was performed on the same set of randomly generated network topologies as used previously. A cell size of 50 meters was used for all tests and the number of boundaries, number of boundary hops, and the total probe byte count were measured. For memory and bandwidth the hello byte count and the boundary record memory size was measured.

6.8.1 Cell Size

The cell size for testing was maintained at 50 meters as defined for 150 meters to check its applicability across radio ranges. As expected there were no boundary probing failures for 150, 175 and 200 meters. However there were failures at 125 meters. This occurred because the optimal radio range for 125 meters is $\sqrt{(125/2)^2/2}$ = 44.2 meters. The cell size for 125 meters was altered to 40 meters and all boundary probing was successful.

6.8.2 Probe Overhead

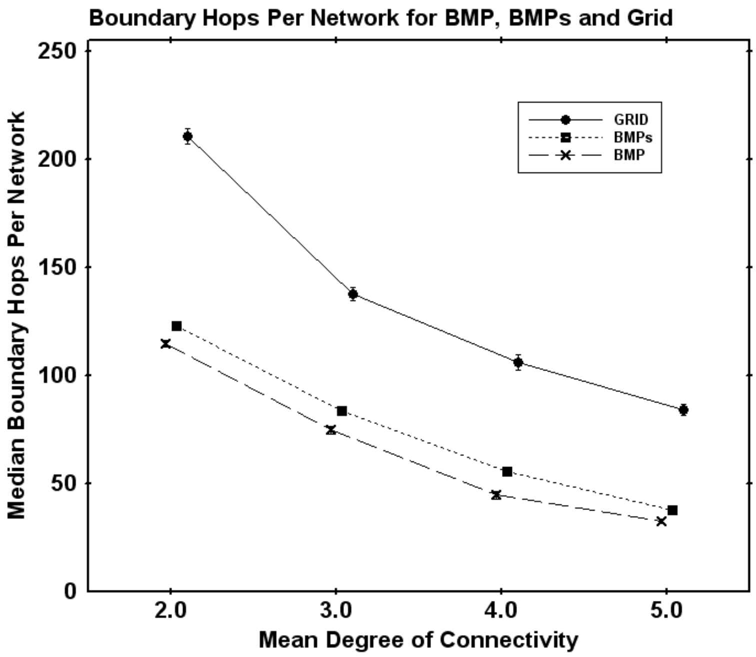

Figure 59: Median boundary hops per network for all probing strategies.

Grid occupancy had a similar number of boundaries as BMPs, because like BMPs it did not use 2 hop neigh information used by BMP. Grid occupancy exhibited a significantly higher boundary size (in hops) as shown in Figure 59. This is expected because both the discovery and confirmation probes for grid occupancy were sent to the closest occupied boundary cell from the current cell which was not the case for BMPs and BMP.

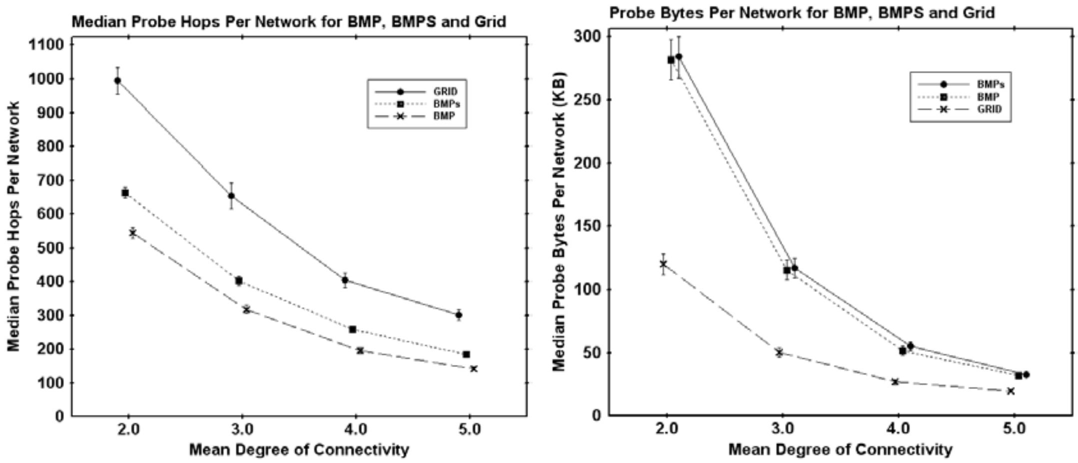

Figure 60: Median of (a) probe hops and (b) probe bytes for all probing strategies.

In Figure 60(a) grid occupancy mapping has a significantly higher probe hop count than both BMPs and BMP, again because both the discovery and confirmation probes are sent to the closest occupied boundary cell from the current cell. However in Figure 60(b) it is clear that although grid occupancy mapping exhibits a higher probe hop count, the actual number of bytes transmitted is significantly lower than both BMPs and BMP.

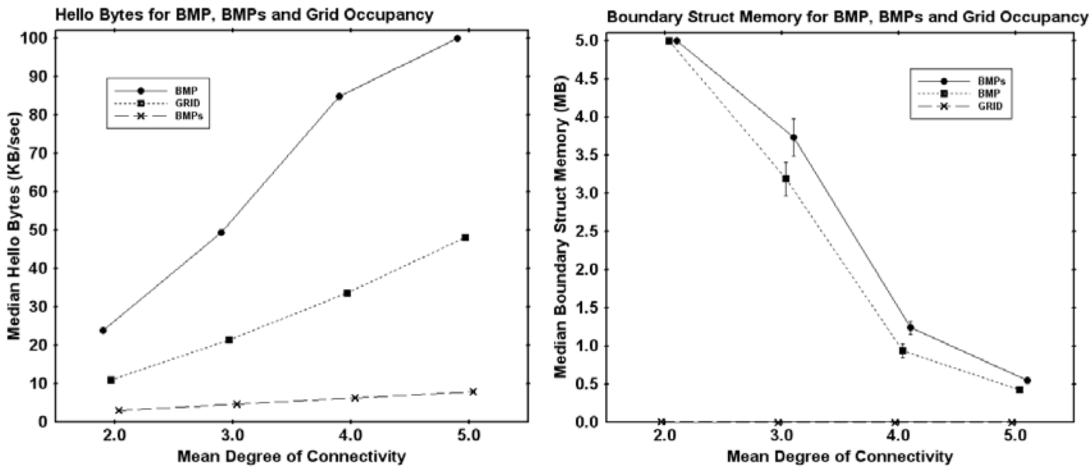

Figure 61: Median (a) hello packet overhead and (b) boundary struct memory overhead for all probing strategies.

Overall grid occupancy mapping performs extremely well in regards to resource usage. Figure 61(a) shows the hello packet overhead for grid occupancy, which has an additional 8 byte extension to each hello message, is still significantly lower than

BMP at all measures of connectivity (BMPs is lower again but in the final considerations it is overly simplistic and not practically usable). This demonstrates that the use of lower resolution information of cell coordinates for adjacency and location in comparison to absolute location (in meters) significantly reduces the bandwidth overhead for hello packets in comparison to BSR, (although this is still a significant cost increase over normal hello messages as used in BMPs).

In Figure 61(a) there is an even more significant improvement in the memory requirements. The memory utilisation for grid occupancy is minimal (visible on the base line of the graph). These values are (left to right) 5.3KB, 3.1KB, 2.0KB, and 1.5KB. There is an additional requirement for the grid structure (in this case 400 cells) but this will be in the order of a few Kilobytes which compared to the approximately 5MB of BMP is negligible.

6.9 Limitations

The occupancy grid map is presented as an alternate low resolution, low overhead system for discovering and distributing global map information regarding voids and the outer boundary. However there are still some areas that need to be refined. There is a need to address the hello message overhead and also to find a better boundary search strategy than the right hand rule. The system proposed in this chapter has not reached the stage of being incorporated into a geographic routing protocol, but the results obtained do show this approach has merit as a mechanism for discovering and distributing low resolution global knowledge. It may also have application in partitioning the system for scalability and because it does not rely on accurate information regarding individual nodes it should be able to better manage its low resolution mapping in dynamic network topologies.

Chapter 7

Experimental Design

> *Application was now made to the Admiralty for experiments to be tried with the compass on board different ships; and the results in five cases being conformable to one of the three laws before deduced*
>
> *Matthew Flinders*
> *A Voyage To Terra Australis Volume I, Preface, 1814*

This chapter will first cover the selection of the dependent and independent variables, identification of the control, discussion of threats to the validity of the experimental process, and selection of appropriate methods of statistical analysis. It will also present the custom network simulation software for modeling and testing of the proposed algorithms.

7.1 Controls

1. The proposed improved forwarding strategy will be compared to the existing forwarding strategies, Greedy forwarding (the most commonly used), Compass forwarding, and MFR.
2. The improved forwarding strategy will be evaluated by substituting it into an existing geographic routing protocol GPSR and comparing its performance to standard GPSR.
3. The proposed boundary mapping protocol will be compared to a basic boundary probing algorithm
4. The proposed geographic routing protocol that uses the improved forwarding strategy plus boundary state information, will be compared to the existing and the improved geographic routing protocol GPSR.

5. The alternate occupancy grid mapping solution will be compared to the previous boundary mapping protocol.

The next section will discuss measures taken to eliminating confounds and ensure experimental validity.

7.2 Dependent Variables

Dependent variable for routing and forwarding

- Looping
  - Percentage of routes which fail due to routing loops (TTL failures)
- Path completion
  - Percentage of routes completed
- Failed route dilation
  - This is the hop count from the source to the fail point plus the shortest path hop count from the fail point to the destination divided by the shortest path hop count from the source to the destination.
- Path efficiency
  - Ratio of actual path length to shortest path length (inverse of route dilation)

Dependent variable for boundaries

- Number of boundaries
  - The number of boundaries probed per network
- Number of boundary hops
  - The total size (in probe hops) of all boundaries per network
- Total probe hops
  - The total number of probe hops including discovery probes, confirmation probes and killed probes per network
- Total probe bytes
  - The total number of probe bytes including discovery probes, confirmation probes and killed probes per network

- Total hello bytes
  - The hello packet bytes transmitted per second per network
- Total boundary struct bytes
  - The total size of data stored in boundary data structs per network

The dependent variables for Hypothesis 1, requiring the comparison of forwarding strategies, were the rate of path completion and the looping. The path efficiency and the failed route dilation were also measured to investigate the cost of improvement in path completion

For Hypothesis 2, requiring the comparison of GPSR and improved GPSR, the dependant variable was path efficiency.

For Hypothesis 3, requiring the comparison of BSR, GPSR and improved GPSR, the dependent variable was path efficiency.

For Hypothesis 4 requiring the comparison of BMP, BMPs, and grid occupancy mapping, the dependent variables were the total probe bytes, the total hello bytes, the total hello packet bytes transmitted per second, and the total boundary struct bytes.

7.3 Independent Variables

The independent variable was the mean connectivity of the network which represented the network density. The values selected were 2.0, 3.0, 4.0 and 5.0 connections per node, representing a realistic range from lightly connected to heavily connected networks. The lightly connected networks represent scenarios which result in lower rates of path completion for the forwarding strategies due to anomalies and local minima, and routing protocols may take excessively long routes due to uninformed decisions regarding the correct choice of direction around a boundary. In contrast a higher node density will have the effect of smoothing indentations in the outer boundary which will reduce or eliminate voids that may be an obstacle to routing.

Network size was not varied (increased) as this needs to be done in conjunction with extensions to the proposed algorithms for parturition of the network for scalability.

7.4 Selection Bias and Sample Size

Tests were performed using 100 randomly generated, fully connected network topologies, each having 100 nodes randomly placed in a 1km square area. Four sets of 100 networks were created with a radio range of 125, 150, 175 and 200 meters with candidate network topologies filtered for a mean connectivity of 2.00, 3.00, 4.00, 5.00+/-0.01 respectively. The 125 meters providing a mean connectivity of 2.0 was chosen because further reduction could not generate fully connected network topologies for the number of nodes and network dimensions. The upper limit of 200 meters with a mean connectivity of 5.0 was chosen because the tests performed used an idealised MAC layer and as a result the protocol performance did not degrade due to congestion. With this limitation the tests performed and the results measured demonstrate the convergence in performance at 200 meters and close to 100% path completion (minimal local minima) for the basic forwarding strategies. The relevance of congestion is important in final application of protocols but does not limit this study in relation to the objectives and goals achieved.

For each protocol being evaluated, tests were performed on the same topologies. This enhances the internal validity but reduces the external validity which is the reason for using a relatively large sample size. The decision regarding sample size needed to consider the complexity of the routing protocols, and the diverse characteristics of sparse network topologies, often resulted in routing failures in as few as 1 in 10,000 routes. Validation of sample size was done by repeatedly performing progressively smaller sampling and evaluating the consistency in graphed means, medians and confidence intervals which confirmed that a sample sizes smaller than 100 networks were not adequately stable.

The static test topologies were generated using a high quality random sampling algorithm [74] to ensure that clustering of nodes did not bias the test results. Individual networks were generated by setting a fixed radio range and randomly placing all nodes without restriction. The networks were then tested for mean connectivity and eliminated if the mean connectivity was not within the specified tolerance for the required value of connectivity.

For testing data packets were sent between all source/destination pairs in all 100 networks for each forwarding strategy using UDP, providing a total of 9,900 source/destination pairs for each protocol at each radio range for each network. Tests were repeated on each test bank for each protocol and measurements were recorded per hop, per route, and per network where appropriate.

Comparison testing was performed between the sample network results with more detailed analysis at route and hop level. The results were found in many instances to have non-normal distributions and therefore results are presented as median values with non-parametric statistical comparisons performed using a Wilcoxon Signed Ranks test for paired samples.

7.5 Selecting a Network Modeling Platform

The following issues were considered when choosing a network simulation platform for modeling and testing of the proposed protocols:

The modeling and testing of a routing protocol ultimately requires the implementation of the routing protocol in a packet switched network simulation environment such as NS2 [75], GloMoSim [76], or OPNET Modeler [77]. These network simulators can be used to model a routing protocol in a real world environment, providing simulation of upper layer functionality, different MAC layers protocols, radio propagation models and mobility scenarios.

The modeling and testing of the proposed protocols was performed using custom software due to the complexity, lack of documentation for installation, lack of tutorials for protocol modeling and testing, and limited documentation of source code, price, and lack of scalability of existing platforms (NS2 [75], GloMoSim [76], or OPNET Modeler [77]) on available hardware at the time of testing.

7.5.1 Custom Network Simulation Software

When modeling and testing began custom software (Figure 62) was developed because existing modeling platforms could not scale sufficiently on the existing hardware available for the project at the time. Although this is not the case at the present time the software created proved ease of protocol implementation making modeling, testing and analysis simple.

The software was developed in C++ using Borland C++Builder. Testing was performed on a Del Latitude 830 with a 2.20GHz Intel Core 2 Duo and 3.5 GB RAM.

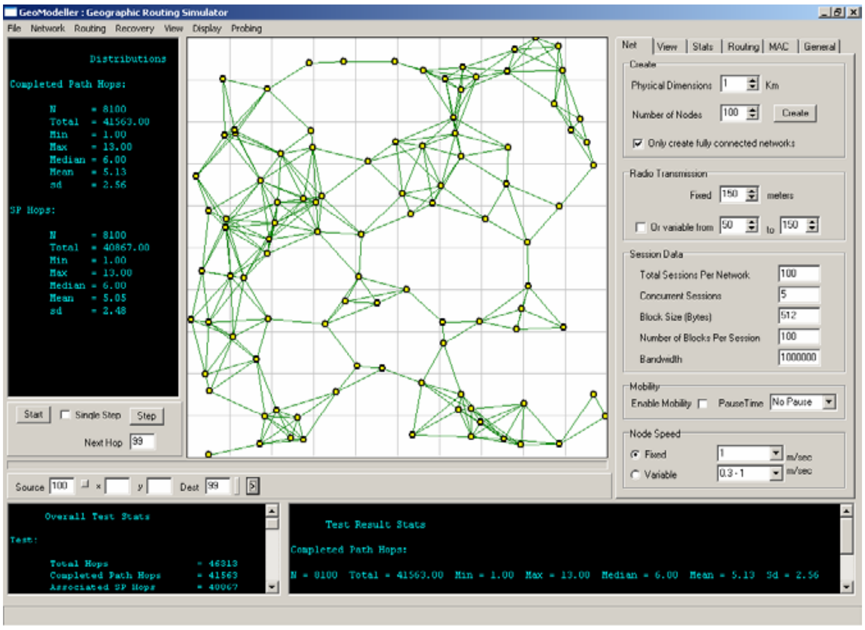

Figure 62: Custom network simulator.

7.5.2 Architecture

Because of the complexity of problem solving in a fully distributed system where data packets, hello packets and probe packets circulate, The software was created without the use of threads to allow the state to be frozen at any instance and any packet followed step by step an any time.

The simulator has been designed with a two tier structure to implement the functionality of a routing protocol as shown in Figure 63.

1. The routing protocol class handles routing decisions and switching between strategies. It also provides the interface call routeQuery(ip_pkt), which is called from the IP layer of the protocol stack and returns the next hop address.
2. A set of associated agents operate independently and supply information to the routing protocol. Agents implement network services and active components of the routing protocol that communicate with adjacent nodes. This allows agents to be used by multiple protocols without duplicating code.

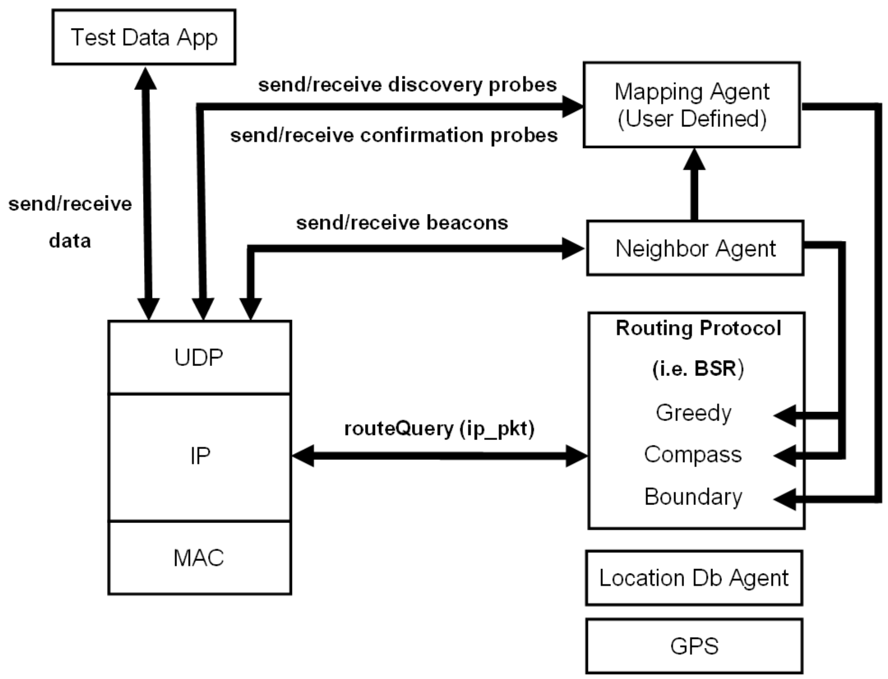

Figure 63: Implementation of routing protocols and agents.

The object structure for each node is:

- GPS class (with mobility)
- Neighbour Agent class
- Location Database Agent class
- Test Data Application class
- UDP protocol stack class
- MAC Layer Emulation class

- Routing Protocol class(es) with a standard inheritable interface and functionality which includes
    - basic forwarding strategies Greedy, MFR and Compass forwarding
    - basic network services: GPS, neighbour agent and location database agent.

7.5.3 Adding a New Routing Protocol

Incorporating a new routing protocol in the simulator requires only a single line to be added to the routing protocol registry class. This makes the protocol available for selection from the Routing Protocol menu and from the Protocol List in the auto test section of the simulator.

The following is an example of adding a protocol to the TProtocolRegistry class.

```
void TRoutingProtocolRegistry::registerProtocols()
{
    ...
    protocols->add(“Your Protocol Name”, new TYourProtocol(param1, param2, ...));
}
```

A new protocol inherits the standard interface from the TRoutingProtocol class which also provides functionality for Greedy, Compass, and MFR forwarding, along with GPS, a neighbour agent and location database agent. Routing protocols and agents are specific to each node when instantiated.

7.5.4 Adding a New Agent

An agent sits at the application layer and provides a network service to the routing protocol. Typically an agent will open a socket to communicate with agents in adjacent nodes. An example is the Neighbour Agent which maintains the adjacency table.

The GPS, Neighbour Agent and Location Agent are available to all routing protocols by default. Users can also define their own agents, then add the new agents by adding a single command to the agent registry in a similar manner to adding a routing protocol. The example below shows the addition of the boundary mapping agent.

```
void TAgentRegistry::registerAgents()
{
    ...
    agents->add("Mapper", new TMapAgent());
}
```

Any registered agent can be activated by any routing protocol (when the routing protocol is selected for use) by adding a statement similar to the one below in the routing protocol init() method.

```
TMapAgent* mapAgent;
...
mapAgent = (TMapAgent*)agents->activate("Mapper");
```

The custom agent interface is then available to the routing protocol.

```
i.e. mapAgent->selectBoundary(destIp);
```

7.5.5 MAC Layer Interface Emulation

The simulator uses a simple generic MAC layer emulation which operates as defined below.

- Accept a transmitted packet
- Drop or forward the packet according to the probability setting for packet loss
- Add latency based on frame size and bandwidth
- Add fixed latency as per settings
- Add variable latency as per settings
- Time stamp the packet and queue
- Release the packet to the receive node according to the time stamp

7.6 Custom Graph Software

To simplify repetitive data analysis during the development phase graph and statistical analysis features (Figure 64) were added to the application for merging extremely large data sets from multiple files, resampling (with the need to re-import data), graphing, and non-parametric statistical analysis. The graphs can be saved or copied to the clipboard for pasting into a document. The graphed values (computed by the graph software from the raw data) are also available in text format for cut and paste into a document.

Without re-importing data the graph can be changed between mean, median, min, max, and count at the click of a button. Confidence intervals (parametric and non parametric) or box plots can also be selected.

Double clicking on any graph point will display the histogram in a side window. The histogram can be transferred in and out of the main window for sampling and copying/saving.

A drop box allows two series to be selected and compared using a paired or unpaired non-parametric test and the results displayed in citation form which can be cut and paste into a document. A spin box allows the user to step along the x axis categories to do a statistical comparison of each pair of data points for the selected series for that category.

Facilities for re-sampling at different sample sizes (without re-importing data) allows for an efficient validation of consistency of results for decreasing sample sizes.

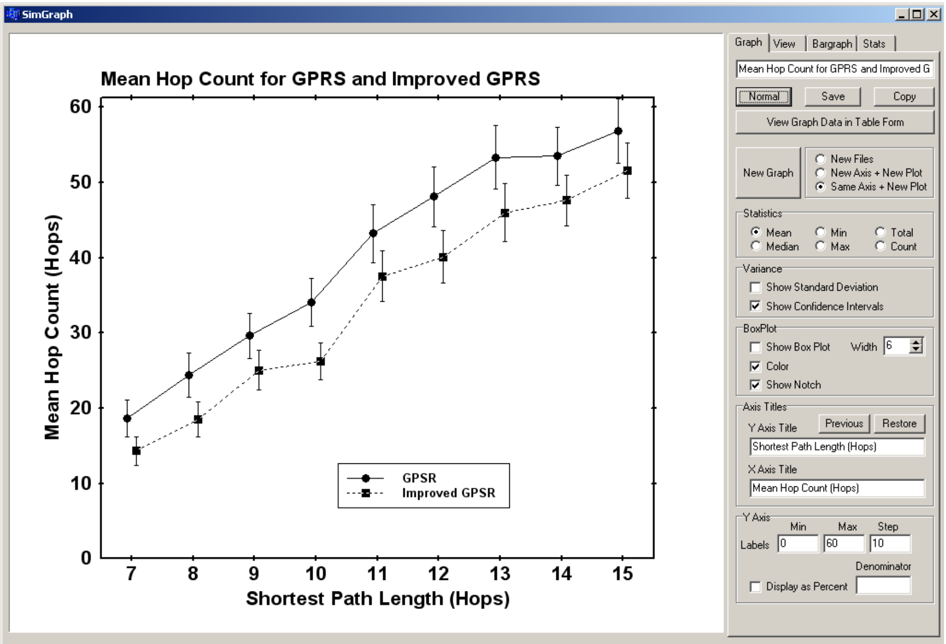

Figure 64: Graph of means with confidence intervals.

7.7 Consideration of Distributions

The distributions of the existing forwarding strategies and the routing protocol GPSR were checked to evaluate the appropriateness of mean or median measure and to determine the most appropriate methods of statistical analysis.

Figure 65 shows the distribution of number of paths completed at each path length for the forwarding strategies Greedy forwarding and Compass forwarding (for the paths completed only and not the failed routes) are normal and skewed to the left. This is relevant to understanding the following distribution of GPSR and other protocols that make uninformed decisions regarding direction of boundary traversal at local minima.

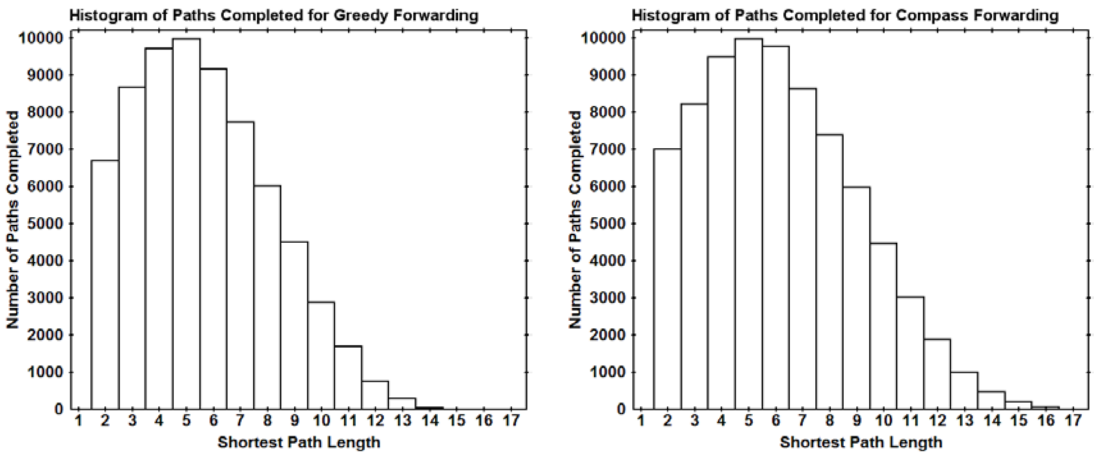

Figure 65: Distribution of number of paths completed at each path length for (a) Greedy forwarding and (b) Compass forwarding.

The problem with GPSR arises because of lack of link state information when switching to an alternate strategy to route packets around a local minimum. This is the reason for the bipolar distribution reported for GPSR in Figure 50 (in chapter 5). The histogram in Figure 66 illustrates this effect for the number of routes completed at each shortest path length from 1 up to 150 hops in a sparse network with a mean connectivity of 2.0 (the worst case scenario).

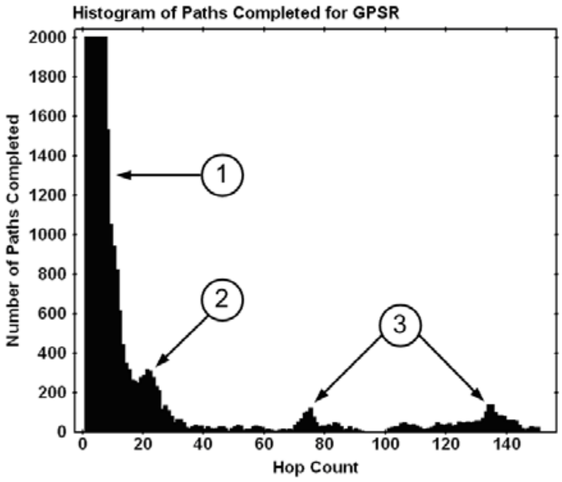

Figure 66: Distribution of paths completed at each route length for GPSR.

The three groups of annotated peaks in the paths completed, displayed against hop count in Figure 66, are interpreted as follows:

1. The basic forwarding strategy Greedy forwarding which handles packets in an efficient manner at the lower end of the distribution in close to optimal path lengths is the dominant peak;
2. The extension of the first peak where the secondary strategy, perimeter mode (planar graph traversal), routes packets around local minima when greedy forwarding fails.
3. The smaller peaks in the mid and upper range of the distribution (above the maximum shortest path length of 35 hops), are the result of incorrect (uninformed) decisions regarding choice of boundary direction in perimeter mode.

These points have also been highlighted in the results and show that parametric statistical tests are not appropriate for the majority of this data.

Chapter 8

Conclusion

> *Matter, rather than manner, was the object of my anxiety; and if the reader shall be satisfied with the selection and arrangement, and not think the information destitute of such interest as might be expected from the subject, the utmost of my hopes will be accomplished.*
>
> *Matthew Flinders*
> *A Voyage To Terra Australis Volume I, Preface, 1814*

This thesis has proposed methods for improving the performance of geographic routing protocols and investigated the cost and benefit of considering a balance between local knowledge and global knowledge. An improved geographic forwarding strategy has been developed, which in its own right has practical application for any existing geographic routing protocol. A boundary mapping protocol BMP and a boundary state routing protocol BSR have also been proposed, along with an alternate boundary mapping approach called occupancy grid mapping. This alternate approach is based upon a low resolution occupancy grid which reduces the overhead for probing and mapping and the dependence on accurate node information. The focus of this research is relevant because there is currently no simple and effective solution to the problem of routing around local minima (which occurs when the basic geographic forwarding strategy fails at a node where there are no neighbours closer to the destination). Strategies previously proposed to address this problem include restricted flooding, backtracking, planar graph conversion using face traversal, depth first search and hybrid approaches that incorporate conventional ad-hoc routing strategies.

The first step towards an effective solution was to investigate the possibility of reducing the incidence of local minima by improving the path completion rate of

Greedy forwarding. This is also driven by the need to forward packets around local minima at nodes having a non reflex angle to isolate local minima to nodes having a reflex angle, making identification of nodes which must initiate boundary problems far simpler. For these requirements, an improved geographic forwarding strategy called Greedy-BoundedCompass forwarding was proposed. This strategy employs BoundedCompass as a fallback strategy on Greedy failure. BoundedCompass forwarding uses Compass forwarding with the constraint that the cumulative angle traversed from the last location closest to the destination does not exceed +/-90^{o}.

For networks with a connectivity of 2.0, 3.0, 4.0, and 5.0 respectively, Greedy-BoundedCompass showed an improvement in median path completion over Greedy of 17.3%, 16.9%, 8.6%, and 3.5% with a decrease of 3.7%, 3.9%, 2.0%, and 0.6% in the median of path efficiency and an increase of 21.2%, 30.4%, 42.3%, and 49.2% in the median of failed route dilation. This decrease in path efficiency was expected as there was no global information available to make an informed decision regarding selection of direction at local minima. These results represented a slight improvement over Greedy-BoundedMFR and a greater improvement over BoundedCompass-greedy and BoundedMFR-Greedy, confirming Hypothesis 1.

The improvement in path completion for Greedy-BoundedCompass produced an improvement in median path efficiency 5.6%, 8.9% 8.4%, and 6.6% for a connectivity of 2.0, 3.0, 4.0, and 5.0 respectively for improved GPSR using Greedy-BoundedCompass in relation to standard GPSR, thus confirming Hypothesis 2.

In addition to the improvements outlined above Greedy-BoundedCompass forwarding also provides the ability to forward packets around boundaries containing non reflex angles. This allows boundaries containing local minima to be easily identified as those containing one or more reflex angles which limits the initiation of boundary probes to those nodes at a reflex angle.

To discover and maintain global boundary state information a simple boundary mapping protocol BMPs was proposed which initiates probes from nodes at a reflex angle. Probes accumulate a list of visited nodes for distribution and use the list to eliminated edge cross over of previously traversed edge to prevent looping. When the packet returns to the source, a confirmation packet containing the boundary list is sent back around the boundary. To deal with more complex overlapping of edges and the complexity that would arise in more realistic networks than those with unit graphs a 2 hop version of BMP was developed which selectively maintains a single intermediate link to potential boundary nodes.

In terms of complexity, the need to search the complete list of accumulated boundary nodes to detect edge crossovers in BMP is not optimal. Also the detection of home was found to be a significant problem as nodes may be visited multiple times and with limited knowledge it is possible to miss detecting the successful circumnavigation of the entire boundary and arriving back at the source (or bypassing the source entirely). Also accumulating and distributing the full boundary node list to all boundary nodes requires excessive bandwidth for probing, and also a large amount of memory for the boundary data structures. In regard to these limitations, development proceeded to enable the evaluation of the benefit of using boundary state information in comparison to local information for routing. The deficiencies in the mapping protocol are addressed later with an alternate mapping strategy.

Under a condition of variable radio ranges of 1:√2, BMP did not exhibit any probing failures due to its ability to accommodate some complexity created by complex edge crossovers. In contrast, BMPs failed to successfully probe all boundaries, but this was expected due to the simple algorithms used to address edge crossovers and looping. Although this does indicate some measure of success, the use of variable radio ranges of 1:√2 is still not realistic in respect to real world applications, although there are some arguments for its relevance.

To test the application of boundary state information for routing, the boundary state routing protocol BSR was proposed. This protocol uses Greedy-BoundedCompass forwarding, and boundary information from BMP to route packets around local minima when Greedy-BoundedCompass forwarding fails. BMP was extended to determine the optimal exit point and direction of traversal based on the shortest path around the boundary, with branches pruned, to determine the next hop selection for routing. For each node in the path BMP re-evaluates the exit point, the direction of traversal, and the possibility of swapping boundaries if the current node is a member of multiple boundaries.

Initially the use of multiple strategies that could move away from the destination caused looping, but this was addressed by applying a strict ordered sequence for the application of each strategy if the proceeding strategy failed, and also restricted the restarting of the sequence to next hop candidates that are closer than the closest ever location.

To test the performance of BSR the path efficiency was compared to the fully distributed geographic routing protocol GPSR. To control for forwarding strategy it was also compared to improved GPSR with Greedy-BoundedCompass forwarding. BSR was found to be significantly better in path efficiency to GPSR and improved GPSR. The improvement in path efficiency of BSR over improved GPSR was 46.1%, 39.4%, 8.5%, and 0.5% at a connectivity of 2.0, 3.0, 4.0, and 5.0 respectively demonstrating the advantage of informed decisions regarding direction of boundary traversal and confirming Hypothesis 3.

Following the favorable results from BSR the probing of boundaries was revisited to formulate an alternate mapping strategy that would address the limitations inherent in BMP in terms of probe bandwidth overhead and the excessive memory requirements. The concept of a low resolution grid occupancy mapping strategy was proposed that does not rely on individual nodes or edges. This strategy represents the network topography as a grid, with each cell represented as empty, locally occupied, remotely occupied. or as a boundary cell. Nodes use a grid occupancy map advertisement mechanism which appends the local trusted occupancy grid to the hello protocol. The issue of optimal cell size is discussed and a window based analysis is presented which deals with the complexity and variability of probe home detection. Testing evaluated probe overhead, additional hello message overhead, and memory overhead for boundary data structures.

For grid occupancy mapping the probe bytes and the hello bytes was in the order of 50% lower than BMP. However the greatest improvement was in the memory requirements which for grid occupancy was in the order of 1.5KB to 5.3KB median per network plus a small additional requirement for the grid structure (400 cells in the example network), In contrast BMP consumed up to 4.9MB per network due to the full path listings and larger data structures and data types.

The occupancy grid map has been shown to be an alternate low resolution, low overhead system for discovering and distributing global map information. This strategy requires additional work to address the hello message overhead and also to find a better boundary search strategy than the right hand rule. However it shows promise due to the fact that it is not dependent on accurate node locations and should be more resilient in dynamic network topologies where nodes are mobile.

8.1 Future Research

Future research will focus on improving the grid occupancy mapping strategy for use with BSR. The main improvements required are:

1. Add boundary route computation to the grid occupancy map to provide next hop selection for BSR
2. Minimisation of hello packet overhead, possibly by only appending local grid occupancy change information to hello messages and only sending on change.
3. Improvement to the probe boundary cell search algorithm that is more robust when the boundary probe moves to intermediate cells that are not directly on the boundary.
4. The capability of probes to step in larger spans along the boundary (as the current algorithm sends probes to the closest occupied cell to ensure reliability).
5. Investigate the application of occupancy grid information to segment the network for scalability, variable radio range, and to limit the range of dependence to minimise the effect of mobility and changes to the network topology.

6. Investigate a scalable local grid addressing scheme centered in the middle that will allow growth, mobility, and scalability.

References

[1] M. Abolhasan, T. Wysocki, and E. Dutkeiwicz, "A review of current on-demand routing protocols," in *Proceedings of the First International Conference on Networking-Part 2*, Colmar, France, 2001, pp. 186-195.

[2] I. Stojmenovic, M. Russell, and B. Vukojevic, "Depth first search and location based localized routing and QoS routing in wireless networks," in *Proceedings of the IEEE International Conference on Parallel Processing,* Toronto, Canada, 2000, pp. 173-180.

[3] T. Camp, "Location Information Services in Mobile Ad Hoc Networks," *Technical Report MCS-03-15,* October 2003.

[4] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 12, pp. 1023-1032, October 2001.

[5] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications,* vol. 32, pp. 246-257, March 1984.

[6] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," in *Proceedings of the IEEE International Conference on Distributed Computing and Systems (Wireless Networks and Mobile Computing Workshop)*, Pheonix, AR, 2001.

[7] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, 2000, pp. 243-254.

[8] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *Proceedings of the 6th international workshop on discrete algorithms and methods for mobile computing and communications*, Atlanta, Georgia, USA, 2002, pp. 24-33.

[9] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial Information in wireless ad hoc networks," *IEEE Personal Communications,* pp. 48-57, February 2001.

[10] L. Blazevic, L.Buttyan, S. Capkun, S. Giordano, J. P. Hubaux, and J.-Y. L. Boudec, "Self-Organization in Mobile Ad-Hoc Networks: The Approach of Terminodes," *IEEE Communications Magazine,* vol. 39, pp. 166-174, June 2001.

[11] S.-C. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," *ACM Wireless Networks,* vol. 7, pp. 513-529, 2001.

[12] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-sequenced Distance-Vector routing (DSDV) for mobile computers," in *Proceedings of the ACM SIGCOMM '94 Conference on Communications Architecture, Protocols and Applications*, London, UK, 1994, pp. 234-244.

[13] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks,* vol. 1, pp. 183-197, October 1996.

[14] T.-W. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks," in *Proceedings of the IEEE International Conference on Communications (ICC),* Atlanta, GA, 1998, pp. 171-175.

[15] G. Chen, M. Gerla, and T.-W. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *Proceedings of the IEEE Conference on Communications (ICC),* New Orleans, LA, 2000, pp. 70-74.

[16] M. S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *ACM/Balzer Wireless Networks,* vol. 1, pp. 61-81, 1995.

[17] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in *Mobile Computing*, T. Imielinsky and H. Korth, Eds.: Kluwer Academic Publishers, 1996, pp. 153-181.

[18] C.-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," in *Proceeding of the International Phoenix Conference on Computers and Communications (IPCCC),* 1996, pp. 480-486.

[19] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, "Signal Stability based Adaptive routing (SSA) for ad-hoc mobile networks," *IEEE Personal Communications,* pp. 36-45, February 1997.

[20] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM),* Kobe, Japan, 1997, vol. 3, pp. 1405-1413.

[21] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, 1999, pp. 90-100.

[22] Z. J. Hass, "A new routing protocol for reconfigurable wireless networks," in *Proceedings of the International Conference on Universal Personal Communications (ICUPC),* San Diego, CA, 1997, pp. 562-566.

[23] C.-C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," in *Proceedings of the IEEE Sensors for Industry Conference (SICON)*, 1997, pp. 197-211.

[24] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications,* vol. 17, pp. 1454-1465, August 1999.

[25] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility," in *Proceedings of ACM/IEEE International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM),* Seatle, WA, 1999, pp. 53-60.

[26] Z. Zhong and T. He, "Achieving range-free localization beyond connectivity," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, California, 2009, pp. 281-294.

[27] T. Watteyne, I. Augé-Blum, M. Dohler, S. Ubéda, and D. Barthel, "Centroid virtual coordinates - A novel near-shortest path routing paradigm," *The International Journal of Computer and Telecommunications Networking archive,* vol. 53, pp. 1697-1711, July 2009.

[28] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in mobile ad hoc networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, USA, 1998, pp. 66-75.

[29] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, TX, 1998, pp. 76-84.

[30] D. S. J. D. Couto and R. Morris, "Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding," *MIT Laboratory for Computer Science, MIT-LCS-TR-824*, 2001 2001.

[31] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)*, New York, NY, USA, 2000, pp. 120-130.

[32] D. Son, A. Helmy, and B. Krishnamachari, "The Effect of Mobility-Induced Location Errors on Geographic Routing in Mobile Ad Hoc and Sensor Networks: Analysis and Improvement Using Mobility Prediction," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 233-245, 2004.

[33] M. Grossglauser and M. Vetterli, "Locating mobile nodes with EASE: learning efficient routes from encounter histories alone," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 457-469, 2006.

[34] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG'99)*, Vancouver, 1999, pp. 51--54.

[35] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," *Technical Report ISI/RR-87-180, ISI*, 1987.

[36] P. Bose and P. Morin, "Online routing in triangulations," 10th International Symposium on Algorithms and Computation (ISACC), 1999, pp. 113-122.

[37] P. Bose, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and J. I. Munro, "Online routing in convex subdivisions," in *International Symposium on Algorithms and Computations (ISAAC)*, 2000, vol. 1741, pp. 47-59.

[38] R. Nelson and L. Kleinrock, "The spatial capacity of a slotted oloha multihop packet radio network with capture," *IEEE Transactions on Communications*, vol. 3, pp. 684-694, June 1984.

[39] J. C. Navas and T. Imielinski, "GeoCast--geographic addressing and routing," in *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Budapest, Hungary, 1997, pp. 66-76.

[40] T. Imielinski and J. C. Navas, "GPS-Based Addressing and Routing," *Request For Comment, Computer Science, Rutgers University*, March 1996.

[41] K. Moaveninejad, W.-Z. Song, and X.-Y. Li, "Robust position-based routing for wireless ad hoc networks," *Ad Hoc Networks*, vol. 3, pp. 546-559, 2005.

[42] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proceedings of the 3rd ACM International Workshop on Discrete Algorithms for Mobile Computing and Communications DIAL M99*, Seattle, USA, 1999, vol. 48-55, pp. 48-55.

[43] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, Los Angeles, CA, USA, 2006, pp. 390--401.

[44] Y.-J. Kim, G. Ramesh, B. Karp, and S. Shenker, "On the pitfalls of geographic face routing," in *Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing*, Cologne, Germany, 2005, pp. 34-43.

[45] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing*, Boston, Massachusetts, 2003, pp. 63-72.

[46] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Annapolis, Maryland, USA, 2003, pp. 267-278.

[47] H. Hwang, I. Hur, and H. Choo, "GOAFR Plus-ABC: geographic routing based on adaptive boundary circle in MANETs," in *Proceedings of the 23rd international conference on Information Networking*, Chiang Mai, Thailand, 2009, pp. 359-361.

[48] B. Leong, S. Mitra, and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information," in *Proceedings of the 13th IEEE International Conference on Network Protocols*, 2005.

[49] L. Barriere, P. Fraigniaud, and L. Narayanan, "Robust position-based routing in wireless Ad Hoc networks with unstable transmission ranges," in *Proceedings of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Rome, Italy, 2001, pp. 19-27.

[50] Y.-J. Kim, G. Ramesh, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, 2005, pp. 217-230.

[51] B. Leong, B. Liskov, and R. Morris, "Geographic routing without planarization," San Jose, CA: USENIX Association, 2006, pp. 25-25.

[52] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, 2003, pp. 1-13.

[53] B. Karp, "Geographic Routing for Wireless Networks," Ph.D. dissertation thesis, Harvard University, 2000.

[54] H. Qingfeng, B. Sangeeta, L. Chenyang, and R. Gruia-Catalin, "FAR: Face-aware routing for mobicast in large-scale sensor networks," in *ACM Transactions on Sensor Networks*, 2005, vol. 1, pp. 240-271.

[55] M. Aissani, A. Mellouk, N. Badache, and B. Saidani, "An efficient real-time routing with presence of concave voids in wireless sensor networks," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, Leipzig, Germany, 2009, pp. 186-190.

[56] L. Blazevic, S. Giordano, and J.-Y. L. Boudec, "A Location-Based Routing Method for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 97-110, March/April 2005.

[57] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, 2005, pp. 230-241.

[58] I. Stojmenovic and S. Datta, "Power and cost aware localized routing with guaranteed delivery in wireless networks," in *Proceedings of the 7th International Symposium on Computers and Communications (ISCC)* 2002, p. 31.

[59] L. Zhang, D. Li, and A. Lim, "Energy-efficient traffic-aware detour trees for geographical routing," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 2, pp. 154-168, January 2010

[60] T.-C. Hou and V. O. K. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 34, pp. 38-44, January 1986.

[61] S. Wu and K. S. Candan, "GPER: Geographic power efficient routing in sensor networks," in *Proceedings of the 12th IEEE International Conference on Network Protocols*, 2004, pp. 161-172.

[62] S. Jung, D. Lee, S. Yoon, J. Shin, Y. Lee, and J. Mo, "A geographic routing protocol utilizing link lifetime and power control for mobile ad hoc networks," in *Proceeding of the 1st ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing*, Hong Kong, China, 2008, pp. 25-32.

[63] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, pp. 70-80, 2002.

[64] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Wireless Networks*, vol. 8, pp. 481-494, 2002.

[65] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for Ad Hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, 2001, pp. 70-84.

[66] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 337-348, 2003.

[67] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 349-365, 2003.

[68] M. Heissenbüttel and T. Braun, "A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks," in *Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN' 03)*, Bern, Switzerland, 2003, pp. 197-209.

[69] B. M. Blum, T. He, S. Son, and J. A. Stankovic, "IGF: A state-free communication protocol for wireless sensor networks," *University of Virginia, Charlotteville USA, CS-2003-11*, 2003.

[70] H. Füßler, J. Widmer, M. Mauve, and H. Hartenstein, "A novel forwarding paradigm for position-based routing (with implicit addressing)," in *IEEE 18th Annual Workshop on Computer Communication Workshop (CCW)*, Taipei, Taiwan, 2003, pp. 194-200.

[71] M. Witt and V. Turau, "BGR: blind geographic routing for sensor networks," in *Intelligent Solutions in Embedded Systems, 2005*, 2005, pp. 51- 61.

[72] Y. Xu and W.-C. Lee, "PSGR: priority-based stateless geo-routing in wireless sensor networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems*, Washington, DC, 2005, p. 680.

[73] H. Kalosha, A. Nayak, S. Rührup, and I. Stojmenovi, "Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks," in *IEEE International Conference on Computer Communications (INFOCOM 2008)*, Phoenix, AZ, 2008, pp. 346-350.

[74] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art Of Scientific Computing*. Cambridge: Cambridge University Press, 1996.

[75] VINT, "The Network Simulator - ns," 2006. Available: http://nsnam.isi.edu/nsnam/index.php/User_Information.

[76] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," *UCLA Computer Science Department Technical Report 990027*, May 1999.

[77] I. OPNET Technologies, "OPNET Modeler Wireless Suite," 2006. Available: http://www.opnet.com/products/modeler/home-1.html.