2006

Dynamic Route Management Strategies for Mobile Ad Hoc Networks.

Susan Rea
Centre for Adaptive Wireless Systems, Department of Electronic Engineering, Cork Institute of Technology, Cork, Ireland.

Follow this and additional works at: https://sword.cit.ie/allthe

Recommended Citation

This Doctoral Thesis is brought to you for free and open access by the Dissertations and Theses at SWORD - South West Open Research Deposit. It has been accepted for inclusion in Theses by an authorized administrator of SWORD - South West Open Research Deposit. For more information, please contact sword@cit.ie.
Dynamic Route Management Strategies for Mobile Ad Hoc Networks

Susan Rea
DYNAMIC ROUTE MANAGEMENT STRATEGIES FOR MOBILE AD HOC NETWORKS

BY

Susan Rea, BEng., MSc., MEng.
Centre for Adaptive Wireless Systems
Department of Electronic Engineering
Cork Institute of Technology

Submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy

Supervisor
Dr. Dirk Pesch

Submitted to Cork Institute of Technology, June 2006
Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material to a substantial extent has been accepted for the award of any other degree or diploma of the university of higher learning, except where due acknowledgement has been made in the text.

Signature of Author:

Certified by:

Date: 14/6/2006
Abstract

A Mobile ad hoc network is a collection of wireless mobile nodes forming a dynamic autonomous network without the aid of a fixed infrastructure. Due to the unpredictable nature of node movement the network topology changes frequently. As a consequence of the limited transmission range of the wireless network interfaces, multiple hops may be necessary for data trafficking between nodes in the network, thus requiring nodes to act as both a router and a host. Ad hoc networks when compared with traditional wireless networks have advantages such as, independence of fixed infrastructure such as access points and base stations, ease of deployment and configuration speed. Such network environments are attractive as spontaneous autonomous network formation is presented, without the need for a centralised system administrator entity.

Exigent challenges exist for mobile ad hoc networks with the issue of efficient routing being a critical concern. A routing protocol must address path generation, path selection, data forwarding and route path maintenance. With the protocol objective being the maximisation of network performance from an application point of view while minimising the routing protocol overhead. Presently, there exist several classes of mobile ad hoc network routing protocols such as, unicast, multicast and hybrid routing. The main contribution of the work presented in this thesis is the development of an overlay architecture for dynamic route management that can be utilised to enhance routing protocol performance for mobile wireless ad hoc networks. The proposed system is designed using fuzzy logic to generate an optimal broadcast flood limiting algorithm that promotes the generation of stable routes and allows the assignment of path weightings with adaptive cache entry expiry times. A reactive distributed policy based congestion control scheme is implemented to alleviate localised network congestion and a cache provisioning scheme for merging network nodes. This is a software solution that promotes distributed decision making while enhancing network performance.
Acknowledgements

And so it is with some relief that I sit down to write my acknowledgements after what seems like never ending months of work. As is often uttered hindsight is wonderful but if I knew what lay before when I began my PhD I think I would have ran in the opposite direction. However, I am truly glad I completed this work and more importantly during my tenure as a research student I have made several wonderful friends that have made for many an intriguing day in the office.

Foremost, I must say thanks to my supervisor Dr. Dirk Pesch for providing me with the opportunity to undertake this work and for directing me through this course of study. In particular, I must say thanks for putting up with me over the last few years.

These acknowledgements would not be complete without saying a big thanks to the visions that are Mary, Olivia and Aisling – Mary for being Mary, a good friend and even better dancer who shares my interest in juniper berries, Olivia for trying to save my soul and running many miles with me, Aisling for being a fabulous way of life, for joining me in many ceramic filled excursions and for proof reading my thesis.

Alan, for also being subjected to proof reading this thesis and various papers, so thanks for all your comments, suggestions, edits - distracting and otherwise! I would also like to thank all my colleagues in the Department of Electronic Engineering who were beleaguered by many questions.

I have to say thanks of course to my family and finally Tony who have been there for me, supporting me over my academic career without ever asking me when I was going to get a real job!
# Table Of Contents

Chapter 1 Introduction ................................................................. 1  
1.1 Motivation & Thesis .......................................................... 2  
1.2 Challenges & Contribution ............................................. 3  
1.3 Thesis Outline ................................................................. 5  

Chapter 2 MANET Routing: Challenges & Solutions ................. 8  
2.1 Introduction ....................................................................... 8  
2.2 Ad Hoc Networks .............................................................. 8  
2.2.1 Current Challenges in Ad Hoc Networking ....................... 9  
2.3 Routing ........................................................................... 12  
2.3.1 Conventional Computer Network Routing Protocols .......... 13  
2.3.1.1 Distance Vector Routing ........................................ 14  
2.3.1.2 Link State Routing .............................................. 15  
2.3.2 Motives for Mobile Wireless Routing .............................. 15  
2.3.3 Routing Techniques for Mobile Wireless Networks .......... 16  
2.4 Mobile Ad Hoc Network Routing Protocols ......................... 17  
2.5 Approaches to MANET Routing Challenges ....................... 18  
2.5.1 Congestion & Load Balancing ....................................... 19  
2.5.1.1 TCP & Mobile Wireless Ad Hoc Networks .................. 20  
2.5.1.2 Route Preference using Congestion Knowledge ............. 22  
2.5.1.3 Cross Layer Congestion Management ........................ 24  
2.5.1.4 Proposed Approach to Congestion Management ........... 26  
2.5.2 Broadcast Limiting & Stable Routing ............................. 27  
2.5.2.1 Broadcast Limiting ............................................. 27  
2.5.2.2 Stable Routing ................................................... 29  
2.5.2.3 Proposed Broadcast Limiting & Stable Routing Strategy . 31  
2.5.3 Cache Management Strategies ..................................... 32  
2.5.3.1 Caching Strategies ............................................. 33  
2.5.3.2 Cache Structure & Capacity ................................... 34  
2.5.3.3 Cache Expiration Policies ..................................... 36  
2.5.3.4 Proposed Cache Management Strategies .................... 38  
2.5.4 Merging Node Awareness ........................................... 40  

Chapter 3 Network Management Architecture ......................... 45  
3.1 Introduction ....................................................................... 45  
3.2 Fuzzy Logic & Communication Networks .......................... 46  
3.2.1 Fuzzy Logic Based Approaches to Network Management . 46  
3.3 Network Management using Policy .................................... 51  
3.3.1 Policy Based Networking as an Approach to Network Management . 51  
3.4 Proposed Network Management Architecture for MANET Nodes .... 53  
3.4.1 Congestion State Monitoring using Policy ...................... 56  
3.4.1.1 Policy Engine ................................................... 56
5.4.1 Aperiodic Cache Provisioning for Merging Network Nodes – an initial investigation .................................................................................................................. 129
5.4.2 Merging Node Cache Provision using Fuzzy Logic ....................................... 134
5.5 Discussion ................................................................................................................... 138

Chapter 6 Conclusions & Future Work............................................................................... 140

6.1 Discussion ................................................................................................................... 141
6.2 Review of Contributions ............................................................................................ 144
6.2.1 Congestion Monitoring ....................................................................................... 144
6.2.2 Fuzzy Logic Assisted Routing ............................................................................ 145
6.2.3 Merging Node Awareness ................................................................................... 146
6.3 Future Work ................................................................................................................ 146

Appendix A Network Management Tools........................................................................... 149
A.1 Fuzzy Logic Set Theory ............................................................................................ 149
A.2 Fuzzy Logic Reasoning System ................................................................................ 149
A.2.1 Fuzzy Inference .................................................................................................. 150
   A.2.1.1 Mamdani Fuzzy Inference .......................................................................... 151
   A.2.1.1.1 Fuzzification ......................................................................................... 151
   A.2.1.1.2 Rule Evaluation .................................................................................... 152
   A.2.1.2 Sugeno Fuzzy Inference ............................................................................ 153
   A.2.2 Composition.................................................................................................... 153
   A.2.3 Defuzzification ................................................................................................ 154
A.2.4 Tuning & System Enhancement ........................................................................ 154
A.3 Policy Based Networking .......................................................................................... 155

Appendix B MANET Routing Protocols............................................................................. 157

B.1 Unicast Topology-Based Routing Protocols .......................................................... 157
   B.1.1 Proactive Routing ............................................................................................... 157
   B.1.1.1 Destination Sequenced Distance Vector - DSDV .................................. 157
   B.1.1.2 Optimized Link State Routing – OLSR ................................................... 159
   B.1.1.3 Topology Broadcast Based on Reverse-Path Forwarding – TBRPF .... 162
       B.1.1.4 Fisheye State Routing – FSR .................................................................. 163
   B.1.1.5 Landmark Routing Protocol (LANMAR) for Large Scale Ad Hoc Networks .................................................................................................................. 164
   B.1.2 Reactive Routing ................................................................................................. 165
   B.1.2.1 Ad Hoc On Demand Distance Vector Routing – AODV ....................... 166
   B.1.2.2 Dynamic Source Routing – DSR .............................................................. 167
   B.1.2.3 Temporally Ordered Routing Algorithm - TORA .................................... 169
   B.1.3 Hybrid Routing ................................................................................................ 170
       B.1.3.1 Zone Routing Protocol - ZRP ................................................................. 171
B.2 Unicast Geographical-Based Routing Protocols ..................................................... 172
   B.2.1 Geographical Routing Algorithm – GRA ...................................................... 172
   B.2.2 Location-Aided Routing – LAR ................................................................. 173
B.3 Multicast Routing Protocols ...................................................................................... 173
   B.3.1 Multicast Ad Hoc On Demand Distance Vector Routing Protocol – MAODV ......................................................................................................................................... 173
       B.3.2 ON Demand Multicast Routing Protocol – ODMRP ................................ 174
B.4 Protocol Performance Evaluation ............................................................................. 174
B.5 Enabling Technologies ................................................................. 176
Bibliography ..................................................................................... 178
Abbreviations & Acronyms

ABR  Associativity-Based Routing protocol
ACK  Acknowledgement
AODV Ad Hoc On Demand Distance Vector
ATM  Asynchronous Transfer Mode
BER  Bit Error Rate
B-ISDN Broadband Integrated Services Digital Network
BNAR_with_NAV Busy Node Avoidance with NAV
BRP  Bordercast Resolution Protocol
CAC  Call Admission Control
CBR  Constant Bit Rate
CDS  Connected Dominating Set
CLR  Clear
CNCL Communication Network Class Library
COG  Centre of Gravity
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance
CTS  Clear-To-Send
CWS  Contention Window
DAD  Duplicate Address Detection
DAG  Directed Acyclic Graph
DBF  Distributed Bellman-Ford Algorithm
DCF  Distributed Coordination Function
DIFS DCF Inter Frame Space
DLAR Dynamic Load-Aware Routing protocol
DSDV Destination Sequenced Distance Vector protocol
DSR  Dynamic Source Routing
DSSS Direct Sequence Spread Spectrum
ELFN Explicit Link Failure Notification
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCAC</td>
<td>Fuzzy Channel Allocation Controller</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FLRS</td>
<td>Fuzzy Logic Reasoning System</td>
</tr>
<tr>
<td>FSR</td>
<td>Fisheye State Routing protocol</td>
</tr>
<tr>
<td>GRA</td>
<td>Geographical Routing Algorithm</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning Service</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communications</td>
</tr>
<tr>
<td>IARP</td>
<td>Intra-zone Routing protocol</td>
</tr>
<tr>
<td>IERP</td>
<td>Interzone Routing Protocol</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGRP</td>
<td>Interior Gateway Routing Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>ISO IS-IS</td>
<td>ISO Intermediate System to Intermediate System</td>
</tr>
<tr>
<td>LANMAR</td>
<td>Landmark Routing Protocol</td>
</tr>
<tr>
<td>LAR</td>
<td>Location-Aided Routing protocol</td>
</tr>
<tr>
<td>LBAR</td>
<td>Load-Balanced Ad Hoc Routing protocol</td>
</tr>
<tr>
<td>LLR</td>
<td>Long Lifetime Routes</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad hoc wireless Network</td>
</tr>
<tr>
<td>MAODV</td>
<td>Multicast Ad Hoc On Demand Distance Vector Routing Protocol</td>
</tr>
<tr>
<td>MBWA</td>
<td>Mobile Broadband Wireless Access</td>
</tr>
<tr>
<td>MCDS</td>
<td>Minimum CDS</td>
</tr>
<tr>
<td>MCL</td>
<td>Minimum Contention Time and Load Balancing routing protocol</td>
</tr>
<tr>
<td>MPDU</td>
<td>MAC Protocol Data Unit</td>
</tr>
<tr>
<td>MPR</td>
<td>Multipoint Relay</td>
</tr>
<tr>
<td>MSSN</td>
<td>Multipoint relay Selector Sequence Number</td>
</tr>
<tr>
<td>MST</td>
<td>Minimum Spanning Tree</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NP</td>
<td>Non-Polynomial</td>
</tr>
<tr>
<td>ODMRP</td>
<td>ON Demand Multicast Routing Protocol</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimised Link State Routing</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PDP</td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point</td>
</tr>
<tr>
<td>PLCP</td>
<td>Physical Layer Convergence Protocol</td>
</tr>
<tr>
<td>PSR</td>
<td>Power-Aware Source Routing protocol</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>QRY</td>
<td>Query packet</td>
</tr>
<tr>
<td>RED</td>
<td>Random Early Detection</td>
</tr>
<tr>
<td>RIP</td>
<td>Routing Information Protocol</td>
</tr>
<tr>
<td>RN</td>
<td>Reportable Node set</td>
</tr>
<tr>
<td>RPF</td>
<td>Reverse Path Forward</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply packet</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request packet</td>
</tr>
<tr>
<td>RRER</td>
<td>Route Error packet</td>
</tr>
<tr>
<td>RT</td>
<td>Reportable subtree</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>RWN</td>
<td>Reconfigurable Wireless Network</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Inter Frame Space</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>SSA</td>
<td>Signal Stability-based Adaptive routing protocol</td>
</tr>
<tr>
<td>TBRPF</td>
<td>Topology Broadcast Based on Reverse-Path Forwarding routing protocol</td>
</tr>
<tr>
<td>TC</td>
<td>Topology Control Packet</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TG</td>
<td>Topology Graph</td>
</tr>
<tr>
<td>TORA</td>
<td>Temporally Ordered Routing Algorithm</td>
</tr>
</tbody>
</table>

vii
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTL</td>
<td>Time To Live</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UDP*</td>
<td>Update packets</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>UWB</td>
<td>Ultra Wide Bandwidth</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
List of Figures

Figure 2.1 Genealogy Of Mobile Ad Hoc Network Routing Protocols .....................19
Figure 3.1 Network Management Architecture For MANET Nodes ..........................55
Figure 3.2 Congestion Monitor Policy Engine .........................................................57
Figure 3.3 Policy Repository State Evaluations ......................................................58
Figure 3.4 Route Discovery/Reply Process ...............................................................62
Figure 3.5 Fuzzy Route Request Limiting And Caching ...........................................63
Figure 3.6 Fuzzy Triangular Membership Function ...............................................64
Figure 4.1 DCF Access Contention .......................................................................81
Figure 4.2 Frame Exchange ....................................................................................81
Figure 4.3 Hidden Terminal Problem ....................................................................82
Figure 4.4 PLCP Protocol Data Unit Frame Format ...............................................83
Figure 4.5 IEEE 802.11b BER vs. SNR .................................................................84
Figure 4.6 Correlating Filter Impulse Response .....................................................86
Figure 4.7 50m x 20m Topology ............................................................................89
Figure 4.8 200m x 200m Topology .......................................................................89
Figure 4.9 500m x 500m Topology .......................................................................89
Figure 4.10 1000m x 1000m Topology .................................................................90
Figure 4.11 50m x 20m Topology Voronoi Diagram .............................................94
Figure 4.12 200m x 200m Topology Voronoi Diagram .........................................94
Figure 4.13 500m x 500m Topology Voronoi Diagram .........................................94
Figure 4.14 1000m x 1000m Topology Voronoi Diagram ....................................95
Figure 4.15 Random Waypoint Mobility Pattern For An Individual Node ..........96
Figure 4.16 The Effect Of Mobility Model Selection On Protocol Performance ....97
Figure 4.17 Simulation Model ..............................................................................99
Figure 5.1 Sample Individual Node Congestion Level during a simulation run .......103
Figure 5.2 Congestion Event Occurrence for varying node density .....................103
Figure 5.3 Average Ratio metrics for individual network nodes – 200 meters squared & constant mobility ..............................................................105
Figure 5.4 Congestion Monitoring Effect on Queue State Metric with constant mobility ..............................................................106
Figure 5.5 Congestion Monitoring Effect on Transmission State Metric with constant mobility ..............................................................107
Figure 5.6 Average Ratio metrics for individual network nodes – 200 meters squared with Pause Times .........................................................................107
Figure 5.7 Congestion Monitoring Effect on Queue State Metric with pause times ..108
Figure 5.8 Congestion Monitoring Effect on Transmission State Metric with pause times ............................................................................109
Figure 5.9 Average Ratio metrics for individual network nodes – 200 meters squared & constant mobility ..............................................................110
Figure 5.10 Link Strength Membership Function ..................................................115
Figure 5.11 Current Route Length Membership Function ............................................... 115
Figure 5.12 Remaining Battery Energy Membership Function ........................................ 116
Figure 5.13 Network Interface Queue Size Membership Function ........................................ 116
Figure 5.14 Fuzzy Broadcast Limiting ............................................................................... 117
Figure 5.15 Route Request Rebroadcast Reduction with fuzzy assisted routing ............ 119
Figure 5.16 Throughput Vs. Cache Capacity with and without fuzzy assisted routing .... 119
Figure 5.17 Average End-To-End Latency Reduction Vs. Cache Capacity with fuzzy assisted routing .............................................................................................................. 120
Figure 5.18 Fuzzy Adaptive Cache Entry Timeout ............................................................ 123
Figure 5.19 Percentage Performance Improvement with Fuzzy Assisted Routing ............. 124
Figure 5.20 Caching Ratio Comparison ............................................................................. 126
Figure 5.21 Throughput Comparison ................................................................................ 126
Figure 5.22 Percentage Reduction in Average End-To-End Delays Measurements With Fuzzy Assisted Routing ........................................................................................................... 127
Figure 5.23 No Cache Provisioning Vs. Cache Provisioning for varying cache intervals with unlimited cache requests .................................................................................................................. 130
Figure 5.24 No Cache Provisioning Vs. Cache Provisioning for varying cache intervals with unlimited cache requests with Cache Provisioning Overhead Included .................. 131
Figure 5.25 Cache Provisioning overhead Vs. varying delay between successive cache requests ................................................................................................................................. 132
Figure 5.26 No Cache Provisioning Route Discovery/Reply overhead Vs. Route Discovery/ Reply with Cache Provisioning .................................................................................................................. 133
Figure 5.27 No Cache Provisioning Route Discovery/Reply overhead Vs. Route Discovery/ Reply with Cache Provisioning overhead included .......................................................................................................................... 133
Figure 5.28 Neighbour Reachability Ratio Membership Function ........................................ 135
Figure 5.29 Congestion Status Membership Function ......................................................... 135
Figure 5.30 Merging Node Simulation Join Time ................................................................. 137
Figure 5.31 Merging Node Awareness Performance Measure ............................................ 137
Figure A.1 Fuzzy Logic Reasoning System ........................................................................ 150
Figure A.2 Fuzzy Rule ......................................................................................................... 151
Figure A.3 Logic Membership Functions ........................................................................... 152
Figure A.4 Degree of Membership Evaluation ................................................................... 152
Figure A.5 Clipping ............................................................................................................... 153
Figure A.6 Policy Engine ....................................................................................................... 155
List of Tables

Table 3-1 Sample Fuzzy Rules..............................................................................................65
Table 4-1 DSSS Physical Layer Characteristics..................................................................83
Table 4-2 Linear Model Battery Parameters..........................................................................88
Table 4-3 Location Point Set Size.........................................................................................93
Table 4-4 Voronoi Diagram Site Set Size.............................................................................94
Table 5-1 Ratio Metric Statistics - 200 meters squared & constant mobility ...................105
Table 5-2 Ratio Metric Statistics - 200 meters squared with Pause Times....................108
Table 5-3 Average End-To-End Reduction.........................................................................109
Table 5-4 Ratio Metric Statistics - 200 meters squared & constant mobility .................111
Table 5-5 Scenario Performance Comparisons...................................................................112
Table 5-6 Cache Entry Weight Assignments & Fuzzy Metrics.......................................122
Table 5-7 Percentage Performance Improvement Achieved with Fuzzy Assisted Routing 128
Table 5-8 Fuzzy Rule Evaluations For $DC$ Consequent Case........................................136
List of Publications

To date, the work presented in this thesis has produced the following publications:


In order to meet the increasing demand for wireless computing and information access in a multitude of environments, more flexible network architectures are required than the existing centralised architectures of cellular mobile communication systems, such as GSM [24]. With the demand for high-speed wireless access to information, radio frequencies in the range of between 2.4 and 60GHz will be employed in the future. At these frequencies, radio transmission experiences severe attenuation and only short range transmission is possible at low power. In order to facilitate a wireless networking environment based on short-range radio communication, current centralised network design concepts are too inflexible and instead the use of mobile ad hoc networks has been proposed.

A mobile ad hoc network (MANET) is a self-organising wireless multihop collection of mobile hosts connected in an arbitrary manner without the support of a fixed networking structure. The ability of network hosts to roam freely lends itself to numerous and erratic changes in topology and consequently communications paths between nodes are not fixed. Also due to the limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another in the network. In order to facilitate multi-hop communication, nodes must provide routing functionality and support the discovery and maintenance of destination routes in a distributed manner. The concept of mobile ad hoc networks has been under development since the 1970s. The definition of standards such as IEEE802.11 [109] in the mid nineties has facilitated the emergence of commercial wireless technology for ad hoc networks and ad hoc network management has been recognised as a demanding progression in mobile wireless technology.

The set of applications for MANETs is diverse, varying from small, static networks to large-scale, mobile, highly dynamic networks. The design of network protocols for these networks is a difficult problem. Irrespective of the application,
Chapter 1 Introduction

MANETs must utilise efficient distributed algorithms when resolving network organisation, link scheduling and routing. The discovery of valid routing paths and the successful delivery of packets in a decentralised environment where network topology is dynamic is not an easy task. Typically, the shortest path (based on a cost function) between a source and a destination in a static network is normally the optimal route. However, this idea is not easily extended to MANETs. Features such as erratic wireless link quality, propagation, shadowing, neighbour interference, limited power and a constantly changing topology must be considered. The network must have the capability to dynamically adjust routing paths to ease any of these effects. The majority of current MANET research assumes IEEE 802.11b to be the underlying enabling technology. However, this is only one of many technologies, others being those such as Bluetooth, ZigBee and ultra wide bandwidth (UWB), which are available for ad hoc networking.

As the field of mobile ad hoc network is still in its developing stage, few real world MANETs have been deployed. Thus, most of the research in this area is based on computer simulation. These simulations rely on several parameters such as the mobility model, traffic patterns, propagation model, and physical layer modelling etc. The accuracy of the simulation model impacts directly on the simulated network performance. Hence, it is essential to use comprehensive models when developing a computer simulation. Computer simulation is also necessary as time and cost restraints can make it impractical to build a real system for performance evaluation. As many real world systems are mathematically intractable they cannot be analytically assessed and make computer simulation necessary for system investigation. Computer simulation is a suitable tool for the modelling and assessment of novel networking concepts, prior to real-world deployment, as simulation facilitates the use of repeatable system scenarios so that the effect of individual network parameters on system performance can be demonstrated.

1.1 Motivation & Thesis

Effective routing is a key issue in ad hoc network environments and has received significant research interest in recent years [2, 16, 17, 112] but there is no one routing
Chapter 1 Introduction

protocol that is suitable for all cases. The current suite of developed routing protocols are normally optimised for a single operating parameter such as energy conservation, power control, shortest path routing, and geographic location awareness. Network conditions, such as traffic load, node mobility and neighbour density, influence routing protocol performance, these conditions are independent of the routing protocol and vary without any control being exerted by the routing protocol at the network layer. For effective route management MANET routing protocols must look towards involving several route metrics and cross layer communications in route construction and control as this will lend itself towards the generation of more stable routes than single parameter optimisation would. Ad hoc networks are also amenable to the application of the principles of emergent behaviour [1] for developing adaptive behaviour as these networks operate in a decentralised manner with nodes exhibiting emergent behaviour as they react to local conditions and affect global performance.

The development of a dynamic multimetric route management framework that can be adapted for use with existing ad hoc network routing protocols is the main motivation for this research work. This thesis does not suggest the development of another new routing protocol but rather a route management framework that acts as an overlay architecture for MANET routing protocols. This overlay architecture aims to enhance protocol performance as routing decisions are based on current network conditions and the local node state. This architecture introduces a distributed local management aspect to node functionality.

1.2 Challenges & Contribution

Efficient routing for mobile ad hoc networks is crucial in enabling optimal network performance and sustainability as the performance of a MANET is strongly correlated with the underlying routing protocol efficiency. The objective of this research work is to develop dynamic route management strategies that can be used in conjunction with routing protocols for mobile ad hoc networks to enhance network performance.

A comprehensive review of recent proposals for routing strategies over mobile ad hoc networks is provided in chapter 2. Despite the plethora of routing protocols available
Chapter 1 Introduction

no individual protocol has identified itself as providing optimal performance for all network conditions. MANET routing is typically optimised for a single parametric, such as traffic load, neighbour interference, channel access times and queuing delays. Such a single metric optimisation has driven the need for a route management strategy that involves multimetric parameter utilisation that facilitates the generation of stable routes with adaptive lifetimes. The use of multiple parameters increases the robustness of the routing protocol as nodes make local decisions based on self and channel observations thus allowing for a more comprehensive interpretation of the network state. The principles of Fuzzy Logic Theory and Policy Based Networking have been used to develop a management architecture for use with ad hoc network routing protocols.

The proposed methods for congestion control routing depend on uncovering lightly loaded nodes, as regards queue loads and channel access times, in the route discovery process. Consequently, high quality paths are generated and are subsequently selected for routing. However, these favoured route paths can become overstrained with traffic demands if constantly used and lead to congested nodes. To alleviate congestion a distributed reactive policy approach is proposed that enables network nodes to periodically determine their congestion status so that nodes that are presently active in routing traffic can, if they identify themselves as being congested or approaching congestion, make their local neighbourhood aware of this. This congestion response awareness leads to traffic being redistributed away from congested network zones allowing the involved nodes to restore their status to a non-congested state. When a node returns to a non-congested state it can again be used for traffic routing. In the research literature broadcast limiting and stable path generation have been treated as separate ad hoc network routing concerns. Presented in this thesis is a combined approach to broadcast limiting and stable routing using fuzzy logic to prevent redundant rebroadcasts and construct stable routes.

Topology based routing protocols typically differentiate between multiple paths for a single destination using the shortest path, in terms of the number of hops, as the best path selection metric. This is not an appropriate method for distinguishing between paths, as the shortest path may not necessarily be the best, as paths should be weighted to indicate their cumulative quality rather than assessing paths solely on their length.
Chapter 1 Introduction

Presented in this work is a fuzzy logic based path weight assignment scheme that uses the path length, received link signal-to-noise ratio, node energy and network interface buffer level as a collective path quality indicator that provides a thorough path selection metric, which enables the routing protocol to dynamically select the best path based on current network conditions.

To purge stale route cache entries a local cache expiry mechanism, that assigns expiry times based on fuzzy logic assessed link characteristics over which information has been received, is proposed. The cache expiry mechanism is independent of tuning parameters and modifies a pre-defined static lifetime to adaptively remove route record entries.

Current routing protocols do not provide for the merging of nodes with an existing network, for network assimilation of merging nodes a route cache provisioning scheme is proposed. This technique furnishes a merging node with the ability to procure copies of their neighbour’s route caches rather than relying on blind flooding for initial topology information. The cache provisioning merging node initialisation method allows existing network nodes to decide, using fuzzy logic, whether to disclose a copy of their route cache. Nodes that proffer cache contents are testifying to their suitability as next hop neighbours. It is necessary that nodes assess their fitness as intermediate traffic routers before tendering route records as nodes that are congested or experience large channel access delays should be avoided in the routing process.

The work presented in this study is evaluated to be a series of beneficial dynamic route management strategies that have been utilised to enhance MANET route management and have been applied to a proactive routing protocol for evaluation purposes, but can also be adapted for use with other classes of routing protocols.

1.3 Thesis Outline

The remainder of this thesis is organised as follows:

- Chapter 2 presents an overview of ad hoc network routing protocols, which discusses conventional routing protocols and how these protocols have been adapted for use in mobile wireless environments. This chapter presents an
Chapter 1 Introduction

overview of several MANET routing protocols and reviews prior research work that addresses current MANET routing issues and derives a motivation for the proposed dynamic route management strategies that are presented in this thesis.

• Chapter 3 discusses the use of fuzzy logic and policy based management in communications networks and describes the proposed management architecture that has been developed using these tools. This chapter concludes with a discussion on how the proposed dynamic route management strategies can be adapted for use with various routing protocols.

• Chapter 4 discusses the need for computer simulation for network modelling and describes the simulation model developed as part of this study. The constituent simulation model entities and parameters are detailed. This chapter also discusses the need for realistic mobility modelling for accurate network simulation and presents the network topology scenarios from which realistic mobility patterns are derived.

• Chapter 5 presents a computer simulation evaluation of the proposed dynamic route management strategies. The effectiveness of the distributed policy congestion monitoring as an approach to congestion management is investigated using average end-to-end data traffic delay measurements in conjunction with queue performance and channel retransmissions. The application of fuzzy logic for broadcast limiting in the route discovery process for the generation of stable routes is investigated. The efficiency of this fuzzy assisted routing scheme is demonstrated using data throughput, control traffic overhead, average end-to-end delay and caching performance as system evaluation metrics. The concept of merging node awareness is proposed as a method of integrating merging nodes with an existing network. The usefulness of this scheme is investigated by examining the effect on route discovery overhead, latency and number of data packets placed directly on a merging node’s queue.

• Chapter 6 provides a summary of the conclusions that can be deduced from the work presented and provides future directions that this research work can take.
Chapter 1 Introduction

- Appendix A describes the theoretical aspects of the network management tools, fuzzy logic theory and policy based management, which are used to implement the proposed architecture.

- Appendix B presents an overview of ad hoc network routing protocols, enabling technologies and briefly discusses performance evaluations of commonly used routing protocols.
MANETs have received significant research interest in recent years as a consequence of the wide availability and economic viability of portable wireless devices. This chapter presents an overview of the related research that has been undertaken in this area. As the work presented in this thesis concerns itself with the development of route management strategies for MANET environments this chapter focuses on MANET routing. Discussed in this chapter are the salient features of ad hoc networks and the challenges faced by MANET architectures. This chapter concludes with an overview of several routing protocols that have been proposed in the literature and reviews prior work that addresses these MANET issues while also highlighting the need for further development in this area with a particular emphasis on routing.

2.2 Ad Hoc Networks

Wireless networking is a technology that, since its emergence, has become increasingly popular in the communications industry as it allows users to access data and services while on the move. There are at present two varieties of mobile wireless networks, infrastructured and infrastructureless. An infrastructured network is one that consists of fixed and wired gateways and base stations. Mobile devices within these networks associate and communicate with the nearest base station. As a mobile terminal moves out of communications range of one base station and towards another, a handover occurs between the previous and current base stations and, as regards user mobility, this is a seamless operation. In contrast, an infrastructureless network or a MANET [2, 112] is a self-organising collection of wireless mobile nodes that form a temporary network without the aid of a fixed networking infrastructure or centralised administration.
Chapter 2 MANET Routing: Challenges & Solutions

As there are no dedicated routers, servers, access points or cabling necessary for ad hoc network deployment MANETs can find numerous applications in areas such as:

- Disaster areas in search and rescue, where fixed infrastructure is destroyed
- Medical application, for instance body area networks where for example, patient vital signs can be automatically monitored and collated.
- Personal area networks for example, for use in mobile commerce applications where users can use portable devices for machine commerce communications such as ticket purchases and bookings.
- Military battlefield zones, for deploying networks in hostile and unknown territories for gathering surveillance information and facilitating communications between remote soldiers using intermediate ad hoc nodes for data relaying.
- Temporary networks in meeting rooms, conferences settings or airport environments etc.
- Vehicle-to-vehicle communications where vehicles enabled with ad hoc wireless technology can from arbitrary MANETs with other road users, possibly for exchange of traffic information so that congested roadways may be avoided or for access to the internet where the vehicles form a wide area wireless network that can communicate with distributed fixed base stations, which provide internet connectivity.

2.2.1 Current Challenges in Ad Hoc Networking

The control and management of MANETs must be distributed among the participating nodes as the network connectivity is time varying. Typically, MANETs are deployed over low power devices with minimum processor capability and small memory mass. Efficient management of MANET environments must contend with issues such as address configuration, channel access, bandwidth allocation, energy conservation and security. Ad hoc networking is a multifarious cross layer problem. The physical layer must adjust to rapidly varying link characteristics. The MAC layer must attempt to support collision free access with fair scheduling whilst trying to deliver traffic over a shared wireless medium with arbitrary links as a consequence of node mobility. Likewise
Chapter 2 MANET Routing: Challenges & Solutions

the network layer must react to a possibly frequently changing dynamic topology and furnish fresh routing information in an efficient manner with minimum bandwidth utilisation. The transport layer must manage capricious packet delays and losses and as such applications should be designed to cope with recurring disconnection and reconnection with peer applications [3].

Presented in [4] is a work that investigates, analyses recent research trends, presents tentative guidelines on MANETs and identifies a series of research interests that are key to developing sustainable MANET environments:

• Routing - efficient routing schemes are necessary to handle fluid topological conditions with varying network sizes and densities. Multicasting and broadcasting should be addressed with location information provision being an issue for geographical based routing. For hierarchical or zone based routing, cluster algorithms must be used to partition networks into geographical zones or service requirement zones.

• Address auto-configuration is essential for autonomous operation of an ad hoc network. IPv4 and IPv6 auto-configuration mechanisms have been proposed and need to be adapted for ad hoc networks. Several methods for duplicate address detection have been proposed and include techniques such as [5] where a host node chooses an address at random and then performs route discovery for that selected address. If a route reply is received then a duplicate address is detected. Duplicate address detection (DAD) is necessary to avoid the misrouting of data traffic. The Strong DAD scheme attempts to detect duplicate addresses within $t$ seconds. Due to network partitions and unknown network size it may not always be possible to estimate a bounded delay for the discovery of duplicate addresses. An alternate scheme, Weak DAD [6], suggests that duplicate address detection be incorporated with route maintenance. Network node IP addresses are tagged with a unique key that forms a pair and this pair is used in all routing related packets. Nodes can have identical IP addresses but are distinguished using these distinct key tags.

• Resource Management - channel access schemes must be optimised for bandwidth utilisation over a shared medium. Busy tone [7] is a channel access
scheme that involves a receiver node transmitting a busy tone, on a separate channel, when receiving data. Neighbouring nodes that hear this busy tone refrain from transmitting and so interference is prevented. The IEEE 802.11b is the most widely adopted channel access scheme for ad hoc networks and is discussed in detail in chapter 5, section 5.4. This mechanism uses transmission initiation packet exchange to reduce channel interference as suggested by [8] and includes acknowledgement packets for reliability [9, 109]. Channel access for IEEE 802.11b is based on a random backoff interval that must be appropriately chosen for efficiency as selecting a small interval can lead to an increased number of collisions as nodes simultaneously attempt to transmit. IEEE 802.11b uses a binary exponential backoff. The selection of this backoff interval is not optimum and other methods suggest using a backoff interval that exponentially increases and linearly decreases [9]. Bandwidth allocation in multihop networks is typically combined with channel access schemes and bandwidth is distributed using the concept of fairness. The Estimation-Based Fair MAC [10] attempts to equalise the throughput/weight ratio for all nodes. Nodes estimate how much bandwidth it can use and the amount of bandwidth used by all other neighbours combined. Proportional Fair Contention Resolution [11] allocates link bandwidth using a probability that is dynamically changed depending on the success/failure at transmitting a packet.

- Energy Awareness - For network longevity power conservation is crucial so nodes should employ power saving schemes. Energy conservation is approached via power saving, where nodes turn off their wireless interface when needed, or power control, where transmit power is reduced. The Power Aware Multi-Access Protocol [12] involves nodes using wakeup channels with nodes using synchronous, asynchronous or hybrid sleeping schedules for power conservation. Power control while promoting energy conservation also attempts to reduce interference. For example, the Power Controlled Multiple Access Algorithm [13] involves nodes transmitting a busy tone at an appropriate power level $P_t$ that allows its neighbour nodes to transmit at a power level less than $P_t$ so that interference is avoided.
Chapter 2 MANET Routing: Challenges & Solutions

- Upper Layer issues – TCP/UDP, the standard transport protocols should be adapted for traffic flows over ad hoc networks. Dynamic IP address assignment must be considered for MANET architectures. For a delay sensitive application such as video streaming, QoS guarantying is required for reliable communications over a shared medium.
- Radio Interface – node-to node transmissions are via the radio interface so optimal antenna design with variable output power is essential in providing communications.
- Security – secure communications is a vital issue for MANET environments to prevent attacks from malicious nodes that can subject the network to bogus traffic flows and IP address spoofing. A wireless medium is easy to snoop on and such an environment is susceptible to malicious nodes. Such nodes can interact with other nodes with the aim of exhausting the nodes batteries. Attacking nodes can flood the network with erroneous routing information so as to make routing inefficient. Zhou [14] proposes the use of digital signatures to protect transmissions and [15] presents a MANET authentication architecture for establishing a certification authority to manage private-public key for message authentication.

Despite the extensive investigation that has been done in these areas, there still exists the need for additional experimentation in further developing a suite of solutions that can address these open MANET research issues. The work presented in this thesis concentrates on MANET routing and as such the remainder of this chapter focuses on routing protocols and route management.

2.3 Routing

Node mobility causes frequent and unpredictable changes in arbitrary ad hoc network topologies and as such, routes in a MANET may consist of many hops through other network hosts. As a consequence of this dynamic topology the design of efficient and reliable routing protocols is an exigent challenge. In ad hoc networks, messages sent by a node may be received simultaneously by all the nodes within its transmission range,
i.e. by its neighbours. Messages requiring a destination outside this local neighbourhood zone must be hopped, e.g. forwarded by these neighbours, which act as routers, to the appropriate target address. As a consequence of node mobility, fixed source/destination paths cannot be maintained for the lifetime of the network. As a result of this, a number of routing protocols have been proposed and developed for wireless ad hoc networks. These protocols have been derived from distance vector and link state techniques, originally proposed for fixed computer networks, and involve determining the best path to a destination in terms of distance or link cost. Such protocols are generally classified as Table Driven, On Demand and Hybrid protocols [2, 16, 112]. At present, the IETF MANET working group [17], set up in 1997, is involved in standardising a number of MANET routing protocols for ad hoc networks.

2.3.1 Conventional Computer Network Routing Protocols

A routing protocol must determine the most efficient path between source and destination nodes so that packets may be sent throughout a network. A routing protocol must address the following issues [2, 112]:

- Scalability – ability to support a large number of nodes & networks
- Ability to adapt to a varying topology & traffic levels in terms of speed and efficiency
- Route discovery
- Route maintenance

Routing protocols can be classified in two categories [112]:

- Centralised or Distributed
- Adaptive or Static

A centralised protocol involves all decisions being made at a central node whereas a distributed one includes all nodes in the routing decision. With an adaptive protocol, route information can be modified as a consequence of network status, such as traffic levels or topology changes, whereas static based protocols update route information at periodic intervals.
Distance vector based routing protocols, such as IGRP [18] and RIP [19], are classed as being decentralised and static and are based on the Distributed Bellman-Ford (DBF) algorithm [125]. This algorithm is used to determine the shortest path (distance metric or number of hops) between a source node and a destination node. In distance vector based routing protocols each node maintains a routing table [2, 16, 112] for its outgoing transmissions that gives the distance metric (cost) from itself to each possible destination node, the destination address and the next node in the path. Each network node periodically broadcasts a copy of its routing table to its neighbours. The nodes then update their routing tables upon receiving these route reports, concerning destination and distance information, from their neighbours. If a destination node is reported that a node does not have in its routing table, it determines the distance metric and adds this destination to its table. Also, if a destination node is reported that a node already has in its routing table but the distance metric is smaller then the routing table is updated with this new information.

Distance vector routing protocols though easy to implement have a number of disadvantages associated with them. The routing table update information is only broadcast at set intervals – if a change in network topology occurs, and consequently in routing information, between broadcast intervals then a node’s routing table may be incorrect and will not be modified until the next broadcast. Making the network broadcasts more frequent will cause the algorithm to converge more quickly, but this will lead to an increase in processor time and routing table transmission bandwidth [112]. The DBF algorithm can also lead to the formation of both long and short lived routing loops [20] as nodes may determine incorrect destination paths using route information that is stale and possibly incorrect due to periodic broadcasting. A routing loop may be formed as follows: consider the nodes \(X, Y, Z\) with \(X\) being connected to \(Z\) via \(Y\). If the link \(Y-Z\) fails, \(Y\) will see that \(X\) is advertising a route to \(Z\) and so \(Y\) will now attempt to route to \(Z\) via \(X\). This will result in the loop \(X-Y-X-Y\)... forming if \(X\) attempts to send to \(Z\). There are however, techniques discussed in [21] that prevent looping.
Chapter 2 MANET Routing: Challenges & Solutions

2.3.1.2 Link State Routing

Link state routing protocols, for instance OSPF [22] and ISO IS-IS [22], are based on Dijkstra's shortest path algorithm [125]. These protocols involve each node sustaining a complete map of the network topology and the cost to reach each destination node. The cost is based on the number of hops, link speed and traffic congestions. Each node monitors the link cost to each of its neighbours and this link cost information is periodically flooded through the entire network. Each node updates their own routing tables accordingly by applying Dijkstra's shortest path algorithm, which determines the shortest path to all destinations from a single source [16]. Loop formation [20] can again be a problem with this type of protocol because of possible incorrect link costs due to long propagation delays of update information through the network.

Distance vector algorithms, based on the Distributed Bellman-Ford algorithm, when compared to link state algorithms are computationally more efficient, easier to implement and require less storage space and network transmission bandwidth. However, distance vector algorithms can lead to loop formation as nodes select their next hops in a distributed manner on the basis of information that is possibly stale and thus incorrect. Protocols based on distance vector must counteract the problem of loop formation.

2.3.2 Motives for Mobile Wireless Routing

Conventional routing protocols, such as those discussed, were designed for static infrastructured network topologies and node mobility was not considered. For mobile nodes in an ad hoc network the periodic transmission of routing table update information requires battery power and as nodes must both send and receive this information it makes conservation of energy difficult. Also, this periodic transmission wastes network bandwidth, as routing table information will often not change from one update to the next. In a static infrastructured network node links are considered bi-directional and of equal quality but this may not be the case for ad hoc networks, thus routes determined by standard protocols may be only unidirectional. Some of these problems have been
approached and solved but conventional protocols as such still remain unsuitable for ad hoc networks [112], however these protocols have been used as a basis for designing MANET routing protocols.

2.3.3 Routing Techniques for Mobile Wireless Networks

Routing is critical in providing high quality end-to-end communications in mobile wireless networks. A routing approach must be aware of the network topography and be able to respond to link breakages quickly so as to prevent degradation of service provision for current communication sessions while utilising the least amount of the available system resources in doing this. The routing scheme must manage node mobility such that communicating nodes are ignorant of it. For static or slow moving networks link changes can be responded to with minimal network resource consumption whereas in highly mobile networks link changes can be so frequent that the routing protocol is unable to cope regardless of the quantity of network resources. In such environments node mobility or link life time predictions are necessary. Other methods such as transmitter power control schemes, error-correcting codes, link layer retransmitting and medium access procedures can be considered for use in conjunction with the routing scheme to enhance the routing procedure.

The routing technique employed in a network depends on the network type and structure. A cellular network consists of a number of cells with each cell being managed by a base station. These cells are used to cover different areas in order to provide radio coverage over a wider area than the area of a single cell. With a multiple cell structure and the ability of nodes to roam freely within these cells, nodes must associate with the corresponding cell base station. This mechanism of mobile devices transferring between base stations is referred to as handover [23] and requires coordination between the mobile device and the base station. Handover typically requires independent multiple access in the existing and new cells, with the mobile device reserving a channel on the new base station. The mobile device will then move from the channel on its present base station to the new channel. GSM [24] handovers for example, involve the mobile device measuring the channel that it is meant to start using before moving over. When the channel is
confirmed okay, the network will direct the mobile device to move to the new channel and simultaneously commence bi-directional communication there, which means there is no interruption in an on-going communications session.

Satellite networks such as Iridium [25], involve a mobile terminal transmitting a signal to one of several satellites that orbit the Earth. Iridium's commercial service was re-introduced on April 1st, 2001 and has 66 Iridium satellites in 6 different polar orbits that provide a global service. Every satellite has a link with four others, two of which are in the same orbit and two of which are in adjacent orbits. At any instant in time there is a minimum of two satellites in contact with a single mobile terminal. The system works by transferring the call to other satellites as the satellite with the initial contact moves out of contact with the mobile device. The signal is then re-bounced back to one of the base stations on Earth and then transferred to a landline to the necessary destination terminal.

Packet-switched networks such as GPRS [23], involve packets that are individually routed between mobile nodes. Packet switching is used to optimise the use of the available network bandwidth and to minimise the latency associated with transmission. Packets are routed to their destination through the best path as determined by the routing algorithm. MANET network routing is based on multi-hop node-to-node routing without the aid of any fixed infrastructure over an arbitrary network topology with network paths having random lifetimes.

2.4 Mobile Ad Hoc Network Routing Protocols

The routing protocols for ad hoc wireless networks have to adapt quickly to frequent and unpredictable topology changes. Routing overhead must be restricted as bandwidth is scarce in a wireless medium and mobile device processing capability is often limited. Protocol scalability is a concern for wireless multihop routing protocols as an increase in network density and mobility can lead to excessive routing overhead. Routing table size is also an issue for MANETs as large routing tables entail large control packet sizes and thus long link utilisation. With a wireless environment and mobile nodes all links should be considered as possibly being unidirectional and a protocol should be able to adapt to this constraint. In terms of battery consumption a protocol must be energy
efficient as the sending/receiving of routing information consumes battery power. Also QoS issues must be regarded, as time-delay and throughput are factors considered by real-time applications. Consequently, the significant characteristics of a MANET environment that an ad hoc routing protocol must consider are [112]:

- Dynamic Topology – Ability to adapt
- Restricted Bandwidth
- Erratic Capacity Link, possibly unidirectional
- Energy Constraints

A significant research effort has been undertaken in developing routing protocols since the inception of MANET architectures. Existing protocols can be classed as follows:

- Unicast Protocols
  - Topology-Based Routing Protocols
    - Proactive
    - Reactive
    - Hybrid
  - Geographical-Based Routing Protocols
- Multicast Routing Protocols

There exists numerous protocols for unicast and multicast communications in MANET environments such as [2, 20, 26, 27, 28, 29, 145], as shown in Figure 2.1, however, there does not exist a single protocol that will satisfy all scenarios, topologies, traffic flows and mobility patterns. Each protocol has its own strengths and weaknesses and as a consequence is suitable for particular environments rather than generic settings. A review of several MANET routing protocols is presented in Appendix B.

2.5 Approaches to MANET Routing Challenges

Routing over ad hoc networks can be significantly improved if routing protocols, such as those described in appendix A, consider issues such as power management, QoS, congestion control/avoidance, load balancing, freshness of routing information, dynamic route entry expiration policies, route selection metrics and broadcast flood limiting techniques. Presented in the research literature are several routing enhancement
techniques that have been incorporated into existing protocols and used to derive new protocols based on existing routing strategies.

Figure 2.1 Genealogy Of Mobile Ad Hoc Network Routing Protocols

2.5.1 Congestion & Load Balancing

Congestion arises in a network when the offered load results in a decrease in the effective network throughput [30]. The fundamental cause of congestion can be
attributed to a device buffer having an associated service rate that is less than its packet arrival rate. This leads to packets incurring long buffer delays at intermediate nodes which in itself can contribute to data source retransmissions and consequent bottlenecks. This resend reaction lends itself to a hastily declining network condition where retransmissions burden traffic flows and the effective throughput can dramatically lessen.

Network congestion is a consequence of network demands outweighing network resources, however approaching the solution to congestion by simply increasing network resources, so that packets are not lost, will not permanently alleviate congestion [31]. Networks with nodes that have limited buffer capacities are prone to packet losses when traffic is high as the buffers can overflow. However nodes with infinite buffer sizes are also predisposed to congestion as packets stored in such a buffer have prohibitively large queue times and would typically have been retransmitted by higher layers. The assumption that the addition of high-speed links to a network, to increase the available bandwidth, will prevent packet losses is not valid as the bandwidth must be effectively managed and shared among competing nodes so that traffic flows are spread over links rather than simply those with the most available bandwidth [30]. Two general solutions have been proposed to alleviate network congestion: congestion control and congestion avoidance [31]. Congestion control is used for controlling the flow of data when congestion has actually occurred in a network whereas congestion avoidance algorithms attempt to predict when congestion will occur and take steps to prevent it occurring. Several congestion solutions have been proposed for ad hoc networks that involve TCP error detection mechanisms, route selection and channel characteristics. These congestion schemes are used for congestion avoidance and control while also selecting the best routes over which to route traffic.

2.5.1.1 TCP & Mobile Wireless Ad Hoc Networks

To prevent or control congestion, a network must transmit data with congestion awareness that can be used to predict or resolve network congestion. Consequently, for MANET environments, robust routing protocols are necessary to manage dynamic topologies with variable wireless characteristics. TCP is typically used as the standard
transport protocol in ad hoc networks, however TCP was designed for wired networks and not for dynamic MANET topographies. For wired networks packet loss and excessive packet delays are indicative of congestion with TCP slowing down its sending rate as a response to this. However with MANET environments packet loss is not solely as a result of congestion, it can be attributed to factors such as neighbour interference leading to high bit error rates (BERs) on the shared wireless channel, variable transmitter powers and dynamic links due to node mobility that can possibly be unidirectional.

To distinguish between route failures and network congestion TCP must be modified to include an error detection mechanism that prevents it from misinterpreting packet loss as congestion. Works such as [32, 33, 34, 35, 36, 37, 38] present TCP handled error detection for use in MANET environments. TCP error detection can be classed as network orientated [33, 34, 35] where nodes require explicit notification to detect congestion or end-to-end [32, 36, 37, 38] with congestion being detected using a metric such as packet round trip time or throughput. A feedback scheme is proposed in [33] that informs the source of a route interruption using a route failure notification that allows a source to suspend its timers and cease sending packets while the destination is unreachable. When the route is re-established the source is informed of this and packet transmission continues. Simulation results were presented to demonstrate the effectiveness of this feedback scheme when compared to a non-feedback based TCP as this method curbs unnecessary transmissions while enhancing throughput.

The authors in [34] propose an explicit link failure notification (ELFN) mechanism that allows nodes to notify the TCP sender. This enables the sender to discriminate between link failure losses and congestion losses with simulation analysis showing that ELFN improves standard TCP. Presented in [35] is a solution to the problem of running TCP in MANET environments that proposes ATCP, which is a layer between IP and TCP that promotes correct TCP performance while ensuring high throughput. End-to-end packet loss detection mechanisms can rely either on a single metric such as packet round trip time [32, 37, 38] or multiple metrics as in [36] which identifies congestion losses using: the delay between consecutive packets, short term packet throughput, packet ordering and packet loss ratio. The results presented for these works [32, 36, 37, 38] all demonstrate improved congestion control performance for
enhanced TCP when compared against basic TCP. Other schemes such as those presented in [39, 40] are unsuitable for use in MANETS as a centralised infrastructure in a wired/wireless network is necessary to successfully extract packet loss information with the wired network being responsible for congestion control in [39] and [40] utilises snoop agents to cache packets that are transmitted between the wired and wireless networks. The purpose of these agents is to retransmit lost packets and prevent the transmission of duplicate acknowledgement packets.

TCP is the standard protocol for reliable data transport in wired networks however for it to be adopted for wireless networks it must be modified to cope with dynamic topologies and variable link characteristics. For TCP to operate in such environments it must successfully differentiate between packets being dropped as a result of link breakages and network congestion. Research, such as that previously discussed, goes towards adopting TCP for use in ad hoc networks.

2.5.1.2 Route Preference using Congestion Knowledge

To distribute or balance network loads over multiple paths congestion information is used to select lightly loaded nodes for routing. The Dynamic Load-Aware Routing (DLAR) protocol [41] considers intermediary node routing loads as the principal route selection parameter. This protocol also observes the congestion state of current active routes and rebuilds the path when nodes that are part of the route have their network interface queue overloaded. Use of the least loaded nodes in route generation distributes and balances network traffic loads. The DLAR route discovery process operates in a similar manner to that of DSR as routes are generated on demand through the use of broadcast floods. As a route request is propagated throughout the network, intermediate nodes append their address and current load, however they are not allowed to proffer route replies as in DSR. After a destination node receives the first route request it waits for a timeout interval for other route requests so that other possible routes can be established. The destination node then selects the least loaded route and returns a route reply over that route. The authors have proposed three algorithms for determining the best route:
Chapter 2 MANET Routing: Challenges & Solutions

- Add the current routing load of intermediate nodes and select the route with the least total load.
- Use the smallest average number of packets queued at intermediate nodes as the route selection metric.
- Use the least number of congested intermediate nodes as the route selection metric, with a node being congested if its current buffer interface queue exceeds some specified threshold value $\tau$.

In the event of multiple routes having equal route selection metrics the route with the least associated delay is preferred. Simulation results exhibit DEAR outperforming DSR, which employs the shortest path in terms of the number of hops as the route selection metric and does not regard intermediate node traffic loads. Both DEAR and DSR generated similar quantities of control traffic but DEAR has a slightly increased data throughput with reduced end-to-end delays.

The Load-Balanced Ad Hoc Routing (LBAR) protocol [42] uses the degree of nodal activity to represent the current load associated with a node. LBAR concentrates on selecting the paths with the least traffic loads, like DEAR, so that data can be routed over paths with the smallest delay. Route discovery has a forward and backward stage, with the forward stage being instigated by a source node broadcasting a setup message to its local neighbourhood. This setup message carries the route cost as determined from the source node to the current node with the cost being updated at intermediate nodes as the route request is propagated through the network. The destination node listens for all route requests for a route-select timeout period and determines the best path, known as the active path, from the available paths using the associated route costs as the selection metric. In determining the cost of a route the degree of nodal activity and traffic interference is used. Nodal activity, $A_i$, is the number of active paths through node $i$, and the traffic interference $TI_i$ is given as:

$$TI_i = \sum_{j} A_j$$

2.1

which is the sum of the activity of neighbouring nodes of node $i$, where $j$ is a neighbour node of $i$. The cost $C_k$ of a route $k$ is:
where \( i \) is an intermediate node on route \( k \) and \( j \) is a neighbouring node of \( i \). Simulation results show the advantages of LBAR over DSR and AODV as LBAR exhibits a higher packet delivery fraction with a lower average end-to-end delay.

Presented in [43] is a congestion avoidance management scheme for use in source routed ad hoc networks. As congestion occurs locally in a network the authors propose that it should be managed locally with traffic being pre-emptively routed around local congested network sites prior to the congestion leading to loss of data and negating the need to modify transmission rates. Nodes are required to observe the network interface buffer load and if over a predefined time interval the average load exceeds a given threshold a congestion message is broadcast to the local neighbourhood. Nodes monitor and record the quantity of traffic they are forwarding over neighbouring nodes and if this volume exceeds a particular threshold the node documents that it has surpassed the forwarding limit for this neighbour node. When a node receives a congestion message from a node that it has exceeded the forwarding limit for, the node must attempt to find an alternate route so as to circumvent the congested node. To demonstrate the effectiveness of the proposed congestion avoidance management scheme it was applied to the DSR protocol and an increase in packet delivery throughput was observed when compared against basic DSR.

### 2.5.1.3 Cross Layer Congestion Management

For MANETs cross layer protocol interactions can be used to improve the routing mechanism. Cross layer design is of significant importance for networks using wireless technologies, given that the state of the physical medium can considerably vary over time. Layers can exchange information so as to make optimal use of the network. Works such as [44, 45, 46] have exploited cross layer protocol interactions and demonstrated enhanced network efficiency.

In [47] the authors propose an ad hoc routing protocol with minimum contention time and load balancing (MCL). MCL uses the medium contention information as the
Chapter 2 MANET Routing: Challenges & Solutions

focal route selection metric. The medium contention information of a node concerns itself with the medium contention access time and traffic load associated with a node. By using the medium contention information in the route selection process MCL lessens latencies and allocates traffic over lightly loaded routes thereby avoiding congested network zones. Simulation results show that, compared with AODV, MCL yields better performance as regards packet delivery ratio, the average end-to-end delay and the normalised routing load.

The authors in [48] suggest monitoring the network interface queue size and the node’s local channel state to extract the congestion state of a node. The MAC layer utilisation level at a node is an indication of how busy or idle a node is. The average MAC layer utilisation is measured as the fraction of time during which a node had packets in its network interface queue awaiting transmission or if a node had attempted to transmit it would have been able to do so. This value is used as a perception of the node’s local channel state. This metric is used in conjunction with the instantaneous network interface queue length as a measure of node congestion that can be used by a routing protocol, at the network layer, to avoid congested network regions. These congestion measurements have been applied to the route discovery phase of the DSR protocol so that lightly loaded nodes in quiet areas of the network are selected for creating source-destination paths for data traffic routing. Simulation results demonstrate the effectiveness of this procedure when compared against basic DSR in terms of improved packet delivery ratio, control traffic overhead and average end-to-end latencies.

Presented in [49] is the busy node avoidance routing protocol with NAV (BNAR_with_NAV). This protocol uses the busy rate, defined as the ratio of the time during which a node is transmitting or receiving to the observation time, as a route selection metric with the intent of diffusing network congestion. For route discovery, broadcast flooding is used with the intended destination listening for all route requests within a specified interval after the first request has been received. The destination node then selects the best path as the one that has lowest cost associated with it. The route cost is defined as the sum of the busy rates of the nodes in a route.
2.5.1.4 Proposed Approach to Congestion Management

The work presented in this thesis does not look at improving TCP for MANETs it instead concentrates on congestion control using the network and MAC layers to implement a congestion management scheme for unicast routing. Congestion aware protocols such as those discussed have concerned themselves with discovering lightly loaded nodes, in terms of queue load and channel access times, in the route discovery process. Thus, good quality paths are constructed and are consequently selected for routing. However, the preferred paths for routing can become over-burdened with traffic if persistently used and this leads to congested nodes. It is proposed that nodes periodically determine their congestion status using a distributed policy approach, so that nodes that are presently active in routing traffic can, if they perceive themselves to be congested or approaching congestion, make their local neighbourhood aware of this. This congestion awareness will cause traffic to be redistributed away from congested nodes allowing these nodes to recover their status to a non-congested state. When a node returns to a non-congested state it can again be used for traffic routing. Current approaches to congestion management look towards building lightly loaded routes but lack the ability to react to congestion once it occurs. The method presented here, irrespective of how routes are discovered and selected, attempts to alleviate congestion by redistributing traffic away from congested network zones.

Nodes will consider three state perceptions when determining their congestion level, network interface queue, channel state, and transmission state. This perception of a node’s congestion state will involve more metrics than those used in [41, 42, 43, 44, 45, 46, 47, 48, 49] and thus will be a more comprehensive interpretation of congestion. Network nodes examine their performance over a period of time rather than taking instantaneous values. The proposed scheme is a reactive approach to congestion control as nodes attempt to divert traffic away from congested network zones and redistribute the load towards lightly loaded network nodes rather than concentrating on constructing lightly loaded paths during the routing protocol path discovery phase. The proposed scheme will enhance existing congestion aware protocols and distribute traffic over lightly loaded network areas thus alleviating congestion and allowing the continued use
Chapter 2 MANET Routing: Challenges & Solutions

of desirable network routes. This congestion monitoring scheme aims to reduce average end-to-end data traffic delays as congested network zones are avoided for routing allowing these congested nodes to recover their status to a non congested state.

2.5.2 Broadcast Limiting & Stable Routing

2.5.2.1 Broadcast Limiting

Broadcasting is a fundamental operating procedure in MANET environments and is typically used for routing protocols [142, 143, 137, 145] in the route discovery phase for propagating route request packets. The most prevalent approach to network wide broadcasting is flooding. A simple flooding scheme for route discovery involves a source node generating a route request packet, which is then sent via a broadcast to the local neighbourhood. This route request packet is then rebroadcast by all receiving nodes, on the premise that nodes do not rebroadcast duplicate packets. While flooding is simple and easy to implement it has been shown in [50, 51] that blind flooding can lead to excessive bandwidth redundancy, contention and collision, a situation referred to as the broadcast storm problem. To lessen the effects of the broadcast storm problem redundant rebroadcasts should be curbed to ease contention and minimise collisions.

The broadcast storm problem is identified and analysed in [50, 51] and the authors have classified broadcasting schemes into the following five categories: probabilistic, counter-based, distance-based, location-based and cluster-based. With a probabilistic scheme network nodes rebroadcast packets according to a certain probability[51]. For a counter-based scheme [50, 51] nodes ascertain whether to rebroadcast a packet by counting the number of packets received during a random delay. Such methods presuppose that when the number of duplicate packets received by a node exceeds a fixed threshold value then the expected supplementary coverage will be negligible thus making rebroadcasting ineffective. A distance-based method [51] decides whether to broadcast based on the relative distance between the receiving and transmitting nodes. In a location-based scheme [50, 51] it is assumed that nodes have positioning capability, such as GPS. A receiving node can accurately determine the additional coverage that will be achieved
Chapter 2 MANET Routing: Challenges & Solutions

by rebroadcasting using geographical locations of network nodes with a specified threshold being used to make the decision whether to rebroadcast. A cluster-based method [51] segregates the network into sets of clusters. Every cluster elects a cluster head and has multiple gateways. The role of the cluster head is to rebroadcast packets to all cluster members and the gateway nodes are used to disseminate broadcast packets to other cluster sets.

A similar work by Williams & Camp [52] classifies broadcasting methods as simple-flooding, probability-based, area-based and neighbour knowledge schemes. For simple-flooding all nodes retransmit broadcast packets. With a probability-based scheme as in [50, 51] nodes rebroadcast with a predefined probability. For an area-based method nodes determine the additional coverage area as in [51] and use this to decide whether to rebroadcast. A neighbour knowledge scheme necessitates nodes maintaining neighbour node information as a means of deciding what nodes rebroadcast packets. Nodes must exchange neighbour information using periodic Hello messages. The interval between these cyclic packets must be chosen so as to prevent collisions and contention while being able to successfully manage mobility.

Tseng et al in [50] have proposed coverage and neighbour knowledge based heuristics in determining the subset of nodes that should rebroadcast packets. Other routing protocols such as [53, 54, 55, 134] build on the neighbour knowledge scheme and use it to limit broadcast floods by curbing retransmissions. Wei and Lu in [53] have developed the use of a connected dominating set (CDS) to create a virtual backbone for routing in MANETs based on work proposed in [56, 57] as a solution to MANET routing. A CDS in a graph G = (V, E) is defined as a set of vertices D ⊆ V such that every vertex not in D is adjacent to at least one vertex in D [126]. A CDS is a minimum CDS (MCDS) if no dominating set D* exists such that |D*| < |D| [57]. Consequently, determining the MCDS of a graph equates to finding the MCDS of a network topology that allows the number of retransmissions to be minimised. Wei and Lu [53] identify the computation of MCDS as being an NP-hard problem and develop a distributed algorithm that effectively approximates the MCDS, with broadcast packets being propagated over the network MCDS. The simulation results presented contrast the performance of MCDS
broadcast limiting with a cluster-based broadcast algorithm for static network topologies with the MCDS based scheme showing a reduction in broadcast redundancy.

Biaz et al [27, 54] propose LAR, which uses a location-based flooding algorithm. Based on the prior location information regarding a destination a source node defines a request zone as a finite search space. Only nodes within the zone are permitted to forward route request packets to their neighbours. The protocol evaluation contrasts the performance of LAR with a basic flooding scheme with the simulation results highlighting a reduction in the number of broadcast control packets being sent over the network for location-based flooding.

Presented in [134] is the concept of multipoint relay (MPR) flooding, which is a distributed two-hop neighbour knowledge flooding scheme that has been used in the OLSR routing protocol [134, 135]. MPR approaches the broadcast storm problem by parsing nodes into groups that are communicated with via relay nodes so rebroadcasting is limited to those relay nodes. Proposed in [55] is the use of a minimum spanning tree (MST) to generate a broadcast limiting tree, where a MST of a graph G is a cyclic connected subgraph of G that uses the minimum total edge length. In evaluating the performance of MST flooding, the authors compare it against MPR based flooding and show that MST flooding is more effective at limiting flooding as energy consumption is reduced and a smaller transmission range is used thus restricting the number of duplicate packets that a node receives.

2.5.2.2 Stable Routing

As flooding is typically used in the MANET route discovery process several paths of varying quality or stability can be generated. However, protocols such as DSR and AODV consider all links as being the same and do not differentiate between paths based on link stability. As an approach to link aware routing protocols like Signal Stability-Based Adaptive (SSA) routing [112] and the Associativity-Based Routing (ABR) protocol [2, 112] have been derived from DSR. The route discovery procedure is similar to that of DSR but both SSA and ABR protocols attempt to select long-lived stable links for routing as a means of reducing link breakages and subsequent route rediscoveries.
These protocols incur additional control traffic overhead when compared to DSR, as periodic beacon messages are necessary to measure the link stability of nodes.

Power Aware Routing as an approach to stability routing is investigated by Singh et al [58], who propose that least cost paths for routing should be determined using transmission power. In [59] power management for MANETs is discussed with the aim of reducing power consumption thus extending battery life and increasing network end-to-end throughput. Routing is based on a minimum power routing algorithm, which is a hop-by-hop shortest path routing strategy with link costs being determined as a consequence of the transmitted power levels as in [58]. Simulation results for power management schemes exhibit increased throughput and reduced power consumption when compared against a scheme that considers all links as being the same. The Power-Aware Source Routing (PSR) protocol [60] modifies DSR to consider battery lifetimes in the shortest path computation. Route requests distributed by PSR include battery information. When the destination receives a route request it waits for a specified time interval after the arrival of the initial request so as to assess all possible paths with the least cost path being selected as the path with the longest battery lifetimes and a route reply is sent over this path.

The hypothesis of longest link lifetimes is investigated as a solution to stability routing in [61]. To lessen the detrimental consequences of link breakages as a result of node mobility, this work proposes to establish routes with long lifetimes rather than the traditional shortest path selection. Protocols that rely on shortest path routing establish routes typically over long distances that may break soon and give rise to additional discovery floods. Increasing the number of hops in a path will reduce link distances but it also increases the risk of route breakage as more nodes are involved and more energy is consumed in transmitting packets over longer routes with increased packet latency also being an issue. The authors propose an algorithm to determine long lifetime routes (LLR) and compare the performance of LLR with shortest path routing and discuss the benefits and tradeoffs associated with long lifetime routing. Obviously, LLR are inherently more stable than shortest path routes as regards packet delivery ratio and packet latency[61]. With LLR, packets can be transmitted with expected delay constraints. However, with shortest path routing link breakages are more frequent causing packet loss and possibly
longer latencies as route maintenance must be invoked to notify nodes of link breaks with route discovery being used to furnish alternate paths. LLR can as stated, consist of more hops than shortest path routes, which increases energy usage and packet delivery latencies. LLR can also contribute to increased network contention and collisions as more nodes are typically involved in the routing process. In order to demonstrate the capability of LLR the authors suggest that LLR be used in suitable applications and environments. A possible application is discovering the shortest routes with longest expected lifetime so that energy consumption and latency may be kept at a minimum. Other suitable applications would be ones that require long session times with low traffic, such as Telnet, as stable routing could be provided via LLR. For QoS-based applications LLR can be used to guarantee QoS levels whereas shortest path algorithms are unsuitable.

2.5.2.3 Proposed Broadcast Limiting & Stable Routing Strategy

Broadcast limiting and stable routing have been treated as separate issues in the research literature. It is proposed here to use a combined approach to broadcast limiting and stable routing using fuzzy logic to inhibit redundant rebroadcasts and generate stable routes. As flooding is typically used in the route discovery process it is intuitive to incorporate a flood limiting scheme to promote stable route generation in the route discovery process. Network resources can be gratuitously expended in broadcast floods that often lead to the generation of unstable paths. Unstable path are classed as paths that have large associated signal to noise levels, consists of low-energy nodes, heavily loaded network interface queues and a high number of hops or paths spread over a large distance between source and destination. To eradicate inappropriate paths from the route discovery/reply process a node rebroadcasts a route request packet if its fuzzy system indicates that it is valid to do so. The merging of multiple metrics into a single decision makes a routing protocol more robust with the aspects of self-management being introduced in the routing process as nodes are aware of their current state, in terms of remaining battery life and network interface queue size, and adapt to the current network conditions. This self awareness in conjunction with the received SNR link value and the current path length enables an informed routing decision, concerning the rebroadcasting
Chapter 2 MANET Routing: Challenges & Solutions

of the route request, to be realised, with the aim of optimising broadcast flooding and constructing stable routes. Fuzzy logic has been selected as the network management tool as it can be applied in a live network with decisions being made based on instantaneous network conditions. Unlike genetic algorithms and neural networks, fuzzy logic does need training data. Policy methods require precise input values for decision evaluation whereas fuzzy systems can cope with varying metric magnitudes.

The proposed fuzzy logic approach could be adopted for use with existing broadcast control algorithms and stable routing methods as previously discussed. For the broadcast control schemes presented in [50, 51], the fuzzy logic system could be applied in determining the suitability of links for cluster formation and used in location-based schemes for selecting the closest neighbours with the best link and node characteristics or with neighbour knowledge schemes for selecting which subset of neighbours should rebroadcast packets. Also, this fuzzy approach could be used in construction of CDS as discussed in [53, 56, 57] for creating a virtual backbone network for routing. The CDS generating algorithm could incorporate fuzzy assessed links in the construction process to produce a stable backbone. Likewise for stable routing schemes [2, 58, 59, 60, 61, 112] this multimetric fuzzy routing scheme can be used to enhance existing protocols that rely on individual metrics for optimisation.

2.5.3 Cache Management Strategies

Reactive routing protocols, such as DSR and AODV, cache recently discovered route information so that routes may be reused thereby preventing route rediscoveries and redundant control traffic overhead. Nodes employ either source route caching or intermediate route caching. Source route caching is where a node has a route to a destination that an application running on itself requests. Intermediate route caching is where a node that is not the necessary destination node replies to a route request from the source with a path to the destination. The purpose of route caching is to make effective use of network topology information by providing a response to route replies and lessening the cost of route discovery in terms of latency and route discovery control traffic overhead. Route caches must furnish fresh routing information as extended storage
of a destination path may leave it obsolete. Stale routing information leads to unsuccessful retransmission attempts and generation of route error notification messages as well as contributing to routing latency. Consequently, efficient cache structure, capacity and expiration policies are needed for cache management. Several Caching optimisations have been proposed in the literature [62, 63, 64, 65, 66, 67, 68, 69, 70, 71] and are discussed here.

2.5.3.1 Caching Strategies

To enhance the beneficial effects as a consequence of source or intermediate route caching the authors in [62] suggest that source routed protocols such as DSR, employ both snooping and tapping as extensions to its caching strategies. Snooping is where a node that is forwarding a packet for some other node, snoops at the unprocessed segment of the complete source-destination path listed in the packet header and extracts from it a route cache entry, which lists the path from itself to the final packet destination. Tapping involves nodes operating their network interfaces in promiscuous mode and allowing the node to process all packets that the interface overhears. This enables nodes to learn of other source routes and route errors that allow a node to purge its cache of stale routing information. These methods were applied to DSR to demonstrate their effectiveness as a cache enhancement strategy with simulation results showing an improved cache performance as a consequence of snooping and tapping. The authors also state that such methods can be applied to other on demand protocols such as AODV, TORA and hybrid protocols like ZRP.

Cache validation is proposed in [63] as a method of ensuring that a cache entry has not expired before a source node uses it for directing data traffic over it. When a node needs to send a packet to some destination and it has an unexpired route cache entry for the destination, the sending node unicasts a probe request packet to the destination node, using the cached route in the packet header as the source route. The sending source node expects one of three possible outcomes:

• It receives a probe reply from the destination that indicates the existence of the route.
Chapter 2 MANET Routing: Challenges & Solutions

- If the cached route is invalid it receives a route error message that indicates the existence of a broken link allowing the source node to purge the route entry and instigate the route discovery procedure.
- If after a specified interval of time the sending node has not received a probe reply it will again initiate route discovery and assumes that some intermediate node is congested and was unable to forward either the probe request or reply before the probe timer expired.

The authors state that such a cache validation technique introduces additional control traffic overhead in the form of probe responses and replies but state that such a drawback is justified given the cost of a node using an invalid cache entry and the simulation results presented illustrate an improved caching performance when applied to DSR.

Presented in [64] is path validation that is a mechanism for establishing path validity and can be likened to cache validation as it also uses probe packets in a similar manner to confirm route existence. The authors state this path validation mechanism can be extended into a cache management policy that needs to determine when and what paths should be validated as validating all paths consecutively can lead to increased overhead. The authors investigated the effect of triggering path validation when a route error is received and state that such an approach restricts the usefulness of path validation as data packets are initially sent over a cached route without validation. This can result in lost data packets and large delays if broken links exist in the cached path. The authors suggest that alternative validation triggering is necessary to fully exploit the effectiveness of this technique and periodic triggering is proposed.

2.5.3.2 Cache Structure & Capacity

Presented in [65, 66, 67] is an analysis of the effects that various route cache designs can have on reactive routing protocols. The work presented in [65] addresses the type of structure necessary to represent a route cache and uses DSR to illustrate the effect of cache structure. The authors propose two types of structure:

- Path cache – this is a structure that stores the complete source destination path and can have multiple entries for a single destination.
Chapter 2 MANET Routing: Challenges & Solutions

- Link cache – alternatively a link cache is one that extracts individual links from route reply packets, which are used to generate a unified graph [125] creating a current topological view of the network.

A path cache structure is easier to implement than a link cache and to find a source-destination path a node simply reads the complete sequence of addresses from the route cache, if it exists. However, for a link cache, a graph search algorithm, such as Dijkstra’s shortest path algorithm[125], is needed to find the destination path. A link cache structure has additional processing overhead when compared with a path cache but a path cache cannot effectively make use of all the possible route information available to it as it merely stores complete paths and does not combine them to produce new paths. The simulation results presented show that a link performs better than a path cache as a higher packet ratio is achieved with lower routing overhead. However, it is also stated that the performance of a path cache can approach that of a link cache with appropriate cache capacity and expiration policy. The work presented in [66, 67] builds on the premise highlighted in [65] that a link cache structure outperforms a path cache structure and use simulation results to confirm this.

The concept of cache capacity is also addressed in [65]. For link caches the maximum capacity is fixed in that a possible $N^2$ links can exist for a network of $N$ nodes. However, for a path cache the maximum storage necessary can be larger and is difficult to define, as complete paths are stored and there is no combining of paths that have common links. This work investigates capacity thresholds for path cache structures ranging from an infinite cache size to fixed cache limits and also proposes the splitting of the cache into a primary and secondary cache. The primary cache stores paths that have been used by the node whereas the secondary cache stores paths that have not been used since being learned. The simulation results presented show that a cache with an infinite capacity performs much worse than the others as regards packet throughput and routing overhead as a consequence of the persistence of stale routing information in the cache. As regards the fixed capacity thresholds, no one scheme outperformed the others when tested over a series of mobility patterns extracted from several mobility models but results confirm the necessity for a fixed cache capacity.
2.5.3.3 Cache Expiration Policies

Cache entries become invalid as a consequence of nodes moving out of transmission range of one another and as nodes are not aware of this there is a need to purge route caches of stale routing information. Cache expiration policies have been used to approach this issue in on demand routing protocols.

Discussed in [66, 67, 68] is the use of an adaptive timer based expiration policy. Presented in [66, 67] is a policy for timing out connections in a link cache structure. The authors investigate a static link lifetime assignment that involves a link being assigned a static duration of $T_1$ seconds that is extended to $T_2$ seconds if the link is used again within the $T_1$ second interval. This policy was used to derive an adaptive link lifetime policy that uses three time-based metrics, born, lastUsed and liveTo to generate a link expiry time. The born attribute indicates the time when a link was added to the cache. The lastUsed attribute is the time stamp when the link was last used to forward a unicast packet and liveTo is the predicted time at which a link expires. Depending on why a link is removed from a cache, either as a consequence of a timeout or a route error, statistical lifetime data are collected. For a route error removal the lifetime of the link $l$ is given as:

$$l = \text{current \_time} - \text{link}[i, j].\text{born}$$  \hspace{1cm} 2.3

and for a timeout $l$ is given by:

$$l = \text{link}[i, j].\text{lastUsed} - \text{link}[i, j].\text{born}$$  \hspace{1cm} 2.4

The estimation of link lifetime, referred to as LIFETIME is initially assigned a static value that is dynamically adjusted, using a first order moving average parameter $\alpha$, whenever a link lifetime $l$ is calculated and is given as:

$$LIFETIME = (1 - \alpha) \times LIFETIME + \alpha \times l$$  \hspace{1cm} 2.5

The simulation results presented show the effect of using varying static lifetime ranges for $(T_1, T_2)$ on protocol performance. The authors state that no one static lifetime range exhibits itself as the best for all test scenarios under various load conditions. However, the results do demonstrate the necessity for adaptive link timeouts and that the proposed link lifetime estimation scheme tracks the optimal link lifetime under various load conditions.
Chapter 2 MANET Routing: Challenges & Solutions

A similar timer based expiry policy is discussed in [68] using a timeout period $\Delta T$ to determine cache entry expiry times. This timeout period $\Delta T$ is updated for cache entries whenever the cached route is used or part of it is overheard being used in unicast packet forwarding. The caching performance is based on the appropriate selection of the timeout period $\Delta T$. A small value of $\Delta T$ can cause the expiry of valid route information. A static $\Delta T$ may be extracted for a given network scenario however, no one $\Delta T$ value is suitable for all network densities, traffic loads and mobility patterns. The adaptive $\Delta T$ value is derived locally at each node based on the average route lifetime and the time between link breaks as observed by the node. The average route lifetime is extracted using the lifetimes of all broken routes in the past. The simulation results confirm the hypothesis that adaptive timeout policies enhance network performance in terms of packet delivery ratio and control traffic overhead.

Proposed in [69] is a link adaptive timeout mechanism based on mobility prediction. It is assumed that nodes are aware of their position, through GPS for example, and that these readings are time stamped so that a velocity and bearing can be determined. The link cache attempts to predict the future topology state of the network using mobility predictions. The authors assume free space propagation with the receiver signal strength being only a function of the distance to the transmitter. Consequently, a link from node $i$ to node $j$ is sustainable while the distance between the nodes is less than some specified transmission range $r$. Each node has an $x$, $y$ position, velocity and bearing. The link expiration time for a link between node $i$ and node $j$ is calculated as a function of these and can be seen as the time taken for the distance between $i$ and $j$ to surpass $r$.

Developed in [70] is a caching scheme suitable for hybrid routing protocols based on nodes monitoring the network topology rather than timer based approach. This caching policy requires cooperation among nodes. A node is set as a cache leader if it writes validated route information to its cache. This node then broadcasts the cached paths to a set of nodes at a distance not exceeding $R$ hops. These sets of nodes are termed cache passive nodes and remove links when they receive an explicit notification from the cache leader node. Cache passive nodes must always have a link to the cache leader node. If such a link fails then the cache passive node removes all route entries that have the cache leader as the next hop. To demonstrate the effectiveness of such a protocol the
authors have modified ZRP to incorporate this cache expiry policy and refer to it as C-ZRP. The simulation results presented contrast the performance of C-ZRP with ZRP and show that C-ZRP requires less overhead for cache maintenance and a lower number of route requests are generated.

Introduced in [71] is the concept of epoch numbers for reducing stale route information in node caches. This strategy is used to prevent nodes from relearning stale routing information regarding a link after having previously heard that this link had broken. Each node sets an epoch number for each new neighbour link detected and for each new link break message that it sends itself. A node’s cache contains the positive (i.e. discovered link) information and negative (i.e. broken link) information along with the corresponding epoch number for each link. Epoch numbers are included in routing packets with nodes trusting packets with higher epoch numbers. Nodes increase their epoch number each time they forward or originate a route request, after it has generated a route error. Nodes will add or expire entries based on epoch numbers. The authors discuss the additional network overhead as a consequence of the use of epoch numbers and conclude that stale routing information does not generally override fresh information when epoch numbering is applied.

2.5.3.4 Proposed Cache Management Strategies

The on demand routing protocol cache management strategies discussed, proposed methods for when and how to cache routing information along with methods to purge stale routing information. However, none of these methods assign weighting to stored cache entries that indicate the quality of the path or individual link. For a routing protocol to function efficiently it should opt for the best route over which to transmit data packets. Hence when caching routes, be it next hop information or whole paths, they should be weighted to signify their quality. Typically, on demand routing protocols such as DSR and AODV use the smallest number of hops as the path selection metric however, this metric is not suggestive of the quality of the path as other routes with an equal number of hops or longer paths may be comprised of higher energy nodes, better link signal to noise ratio (SNR) or queue levels and so would be a better path over which
to route packets. It is proposed to use a fuzzy logic system to assign weights to a path so as to discriminate between paths using the fuzzy metrics: path length, link SNR, node battery energy and present network interface queue level. The fuzzy path weighting provides a thorough path selection metric rather than simply using the shortest path and enables the routing protocol to dynamically select the best path based on present network conditions.

The cache expiration mechanisms presented in [66, 67, 68] are adaptive timer based policies, and in [69] mobility prediction is used to predict link breakages and consequently expire routes. In [70] node cooperation is necessary to implement cache entry expiry and finally in [71] epoch numbers are used as a means of detecting stale routing information. These methods do not consider the characteristics of the link over which packets are received when assigning expiry times or determining link lifetimes.

The cache expiry method proposed in [66, 67] uses a range of static lifetime assignments and from that extract the optimal range for selecting a specific range to adaptively assign link expiry times. This method works well for the given network scenarios and node density however this is not a generic mechanism that can be applied to all network topologies and traffic loads as the range for selecting the adaptive timeout would need to be tuned for each scenario. Likewise in [68] a tuning parameter is also used so due consideration must be given when selecting a value for this tuning parameter as an incorrect choice may skew results. The mobility prediction used in [69] to predict link breakages is based on free space propagation and assumes received signal strength is a function of distance only. For a thorough investigation of the effectiveness of this scheme shadowing should be incorporated into the propagation model for realistic mobility predictions and this scheme also assumes nodes possess GPS capability.

Presented in [70] is a cache expiry mechanism that requires cooperation among nodes. Subsets of nodes rely on cache leader nodes to direct cache recording and deletion, consequently, nodes do not make local caching decisions. Epoch numbers are used in [71] to ensure cache freshness. This method, as in [70] involves nodes depending on other nodes to properly direct caching. In this case nodes must manage epoch numbers so that stale routing information can be purged from node caches. Mechanisms requiring node cooperation and coordination are susceptible to attacks from malicious nodes, which
can generate spurious caching directions or generate epoch numbers that are out of synchronisation for the scheme presented in [71].

It is proposed to develop a novel fuzzy logic based local cache expiry mechanism that assigns expiry times based on multiple link characteristics over which information has been received. This policy does not necessitate the need for cooperation with other nodes and a tuning parameter is not needed when determining link lifetimes. The proposed cache expiry policy modifies a static expiry time by increasing or decreasing it based on the quality of the link that is being cached. This expiry method is more intuitive than the others discussed as it considers the quality of the actual link whereas the other mechanisms consider all links as being the same and do not consider link quality when calculating expiry times or TTL values.

2.5.4 Merging Node Awareness

The issue of cache provisioning for merging nodes joining an existing network is not addressed in the research literature, as simulated ad hoc networks are considered to have a set number of nodes at simulation start time with this node number remaining static for the experiment lifetime. For realistic network scenarios this is not a valid assumption as nodes can enter or leave a network at will. Route maintenance techniques have been incorporated in both proactive and reactive routing protocols to manage broken links that arise as a consequence of node mobility and interference. Such techniques can also be assumed to administer the departure of existing nodes. However, for the network integration of merging nodes no such techniques exist.

Both proactive and reactive routing protocols for mobile ad hoc networks exploit route caching and route table strategies in an attempt to minimise control traffic generation. Route cache and route table management must be efficient and dynamically adapt to continuing topology changes. A node once having assembled a cache or route table can commence data transmission. However, merging network nodes can arbitrarily join a network and will initially need to build a routing table/cache for transmission of self-generated data traffic as such nodes are unaware of the present network topology. Consequently, merging nodes can contribute greatly to the overall control traffic
Chapter 2 MANET Routing: Challenges & Solutions

overhead. A cache provisioning scheme is proposed for merging nodes that is applicable to reactive routing protocols but it could also be extended to proactive routing protocols. A merging node should extract valid routing information from their local neighbourhood by soliciting copies of its neighbour's route caches rather than relying on blind flooding for initial topology information. However, during this node initialisation process due consideration must be given to the impact of the cache provisioning scheme on the nodes that are required to provide copies of their routing tables. Such nodes are promoting the routing of data traffic via themselves, which can lead to or enhance network congestion. The concept of cache provisioning has previously been employed in other routing protocols, such as DSDV [20]. However, DSDV functionality relies on the periodic exchange of routing information, which involves each node periodically broadcasting to its local neighbourhood a copy of its route cache. This periodic route trade in an ad hoc network has a number of disadvantages - in particular it contributes to control traffic and may often be unnecessary as existing route records may still be valid. In addition, periodic exchanges drain battery power as they flood the network, and nodes forward route information unnecessarily. The principle presented here is to provide route information to a new network node in an aperiodic and controlled manner so as to reduce control traffic bursts associated with new nodes having non-existent caches. The proposed scheme allows existing network nodes to decide, using fuzzy logic, whether to divulge a copy of their routing cache thereby keeping congestion to a minimum.

2.6 Conclusion

The performance of an ad hoc network is directly correlated with the underlying routing protocol efficiency. Several protocols have been proposed for MANET routing however no individual protocol has identified itself as providing optimal performance for all network conditions. Routing over MANETs is implemented as a distributed local process at the network layer with nodes making local decisions that affect global performance. A routing protocol must dynamically adapt to local network conditions that are determined by network layer independent factors such as relative node mobility rates, traffic flows and immediate neighbour node connectivity. Control traffic overhead can be
proportionally related to relative node mobility [155, 72], with an increased mobility rate link breakages occur at a greater rate and control traffic is generated as a response to manage these link breakages. Reactive protocols, such as DSR and AODV, are made aware of link breakages when a node fails to receive an explicit acknowledgment for a unicast packet for an active communications session. Proactive protocols, like OLSR and FSR, are impervious to effects of mobility as periodic topological updates are broadcast to maintain neighbour connectivity. Traffic flow has an effect similar to mobility rate [72] for reactive routing based protocols as control overhead is driven by the number of active communication sessions. In contrast, proactive routing protocols generate periodic control overhead irrespective of the current network load, but a full topological view of network connectivity is readily available. Network size and node density also impacts on protocol performance, with a dense network and a relatively small physical area one hop neighbour connectivity is high, which can lead to increased channel access times, collisions and retransmissions. For sparse networks, nodes must rely on a relatively small subset of neighbour nodes for data forwarding. This can cause congested network zones with traffic flows being concentrated on a small number of nodes. For larger network sizes a routing protocol must be scalable with hierarchical protocols, such as ZRP, being best suited as nodes form clusters for interzone and intrazone routing.

This chapter has discussed approaches for MANET routing and has shown the disparity of the approaches towards MANET routing. Several optimisations, proposed in the research literature, for enhancing routing performance have been discussed. From this discussion it is evident that MANET routing is typically optimised for a single parametric, such as traffic load, neighbour interference, channel access times and queuing delays. This single metric optimisation has highlighted the need for a route management strategy that relies on multimetric parameter utilisation that can aid in the generation of stable routes with dynamic lifetimes. The use of multiple parameters increases the robustness of the routing protocol as nodes make local decisions based on self and channel observations thus allowing for a more comprehensive interpretation of the network state.

To support MANET routing a series of dynamic route management strategies have been proposed that look towards compensating for the weakness of a single route
selection metric. These methods intend to complement a node's routing efficiency. A routing protocol based at the network layer has no influence over the traffic that is generated at the application layer and nor does it exercise any control over node movement, it must adapt to dynamic network conditions as results of the influence of these factors. The proposed route management strategies rely on multimetric observations that are used in making local routing decisions. Existing routing protocols and algorithms implement congestion avoidance schemes in conjunction with route discovery as approaches to congestion management. The method proposed here is a reactive congestion control technique that alleviates congestion after it has occurred. The premise for the development of such an approach lies in the fact that if a routing protocol has constructed route paths over the least congested nodes available, these paths are used for data forwarding and can, if extensively used, become congested. To alleviate this congestion the proposed congestion management method will run in parallel with the routing protocol and attempt to divert network traffic away from congested nodes and to establish alternate paths for data routing.

Reactive routing protocols employ route tables or caches to record active route paths and use either the shortest path or most recently used path as the route selection metric. Such metrics are not indicative of path quality as they consider all links/paths as being of the same quality. It is proposed here to use fuzzy logic to assign a path weighting to represent path quality. This rating considers link SNR, path length, node energy and node load. Such a weighting is indicative of the quality of a route and allows nodes to differentiate between multiple paths for a single destination. For a single route selection metric it is not feasible to determine the best available path, DSR considers the shortest number of hops as its route selection metric so for paths of equal length it is not possible to determine which is the best.

The cache expiration policies previously presented rely on tuneable parameters, mobility predictions or node cooperation for cache entry expiration. As an alternative, a simple adaptive cache record expiry policy is proposed that uses the path weighting metrics to determine dynamic route lifetimes for cache entries. This method, unlike those previously discussed, does not consider all links as being uniform and determines
Chapter 2 MANET Routing: Challenges & Solutions

dynamic route lifetimes for cache entries. This cache weighting and deletion strategy can be incorporated in the route discovery phase of a reactive routing protocol.

To restrict broadcast floods and promote the generation of stable routes in route discovery a combined fuzzy logic based broadcast limiting and stable routing approach has been proposed. To preserve scarce network resources and restrict floods to stable routes it is proposed to use a fuzzy logic based decision strategy that determines whether to rebroadcast route discovery packets.

A novel method for the integration for the merging of nodes with an existing network has been proposed. This approach provides merging nodes with current routing information and aims to lessen the control traffic overhead associated with merging nodes establishing initial route records. Existing network nodes assess their suitability as next hop neighbours, using fuzzy logic, and based on this decide whether to disclose cache records. Merging nodes extract the best available subset of records from all route information received.
Chapter 3 Network Management Architecture

3.1 Introduction

The principles of Fuzzy Logic Theory [73, 74] and Policy Based Networking[75] have been used to develop a management architecture for use with ad hoc network routing protocols. The proposed dynamic route management strategies are to supplement existing MANET routing protocols by incorporating multimetric observations for channel awareness and node-self awareness. As a routing protocol must adapt to dynamic network conditions both the channel and node observations are used to interpret the current network state and are used in conjunction with the network layer routing protocol to influence local routing decisions. By monitoring its local channel a node can determine the channel activity and the quality of the links within its one hop neighbour set. Likewise, by assessing its own state a node can infer its suitability as a router for its local neighbourhood. Fuzzy logic has been used to develop:

- A broadcast flood limiting algorithm, which can be applied to the route discovery phase of a MANET routing protocol, to restrain the propagation of route request packets and to promote the generation of stable routes
- A path weighting policy that is indicative of route quality, which is used for route cache path differentiation and dynamic route lifetime assignment
- A merging node awareness algorithm that provides nodes joining network with active route information, which enables these merging nodes to route over the best subset of available one hop neighbours.

Policy based management is used to implement a congestion control policy that diverts traffic away from congested network zones. Discussed in Appendix A are the theoretical aspects of fuzzy logic theory and policy based management. The remainder of this chapter discusses the use of these tools in communications networks and describes the proposed management architecture that has been developed using these tools.
Chapter 3 Network Management Architecture

3.2 Fuzzy Logic & Communication Networks

The principle of fuzzy logic set theory is to furnish a framework for approximate reasoning that can deal with real world imprecision and uncertainty [73]. Present day communication networks are complex and dynamic. Systems resource allocation, routing, QoS issues, fixed buffer capacity, queuing, access, congestion control and variable traffic flows contribute to unpredictable network performance. With an escalating need for reliable communication services, network management techniques are essential for providing guaranteed communications with fair network resource allocation and use. The use of Fuzzy logic has been proposed to address these management issues in communication networks such as ATM networks for queue management [76], resource allocation [77, 78, 79], QoS [80, 81], call admission control [82, 94, 95], congestion management [83, 84, 85, 86, 87] and to lesser extent in routing over Ad Hoc networks [88, 89, 90, 91, 92].

3.2.1 Fuzzy Logic Based Approaches to Network Management

Fuzzy logic is typically applied to control engineering problems however, recently it has been successfully applied to the management of communications networks. Fishwick [93] discussed the use of fuzzy set theory as a means of encoding uncertain or qualitative simulation knowledge. Fuzzy queue management has been proposed in several works, such as Bonde and Ghosh [76], who have utilised the concept of cell blocking using a fuzzy threshold function that intentionally drops a percentage of incoming cells from other switches. The work investigates the theory of accepting or rejecting incoming cells. Comparative simulation results with a binary threshold scheme demonstrate the effectiveness of the fuzzy queue management in adapting to dynamic network conditions as the fuzzy threshold queue management scheme causes the buffers to exhibit ‘soft’ behaviour facilitating an improved aptitude toward managing sharp condition changes in the network thereby reducing cell drop rates and thus increasing cell throughput. Presented in [77] is a survey paper that examines the use of computational intelligence applications, including fuzzy logic, in ATM network control. Barolli et al
Chapter 3 Network Management Architecture

[78] have presented an intelligent fuzzy routing scheme for improving ATM network performance through the use of tagged cells to maximise the utilisation of available resources. For traffic management in high-speed networks the authors in [79] have developed a fuzzy logic capacity estimator for bandwidth allocation with simulation results showing that a combination of such an estimator with existing techniques improved estimations. Vasilakos et al [80] have used evolutionary-fuzzy prediction techniques in interdomain routing of broadband network connections with QoS requirements. The traffic load on individual links is predicted and this is used to determine the shadow cost of selecting paths allowing the selection of the best candidate path so that network demand can be satisfied. Lo et al [81] have also addressed QoS guarantying and have proposed a fuzzy channel allocation controller (FCAC). The FCAC consists of a layer architecture that consists of a fuzzy admission threshold estimator and a fuzzy channel allocator with the channel allocation being determined based on the admission threshold. The simulation results presented show that the FCAC can always guarantee the QoS requirement of handoff failure probability for all traffic loads while improving system utilisation and reducing handoff rate when compared with the combined channel allocation scheme [81].

Fuzzy based admission control has been presented in numerous works such as [82, 94, 95]. The authors in [82] propose a fuzzy call admission control (CAC) scheme that makes a fuzzy decision, using the mobile mobility information, multimedia traffic characterisation, QoS provisioning and system resource utilisation, to accepts a mobile station service request if the QoS requirements of all the currently active calls in the target and neighbouring cells can be guaranteed in addition with the QoS level for the new call. Simulation results presented demonstrate the ability of the proposed fuzzy CAC scheme to: offer low new call blocking and handoff call dropping probabilities, satisfy the outage probability requirement while making efficient use of the system resources. Presented in [94] is a framework for the integration of a fuzzy logic tool for admission control in queuing systems. The proposed tool is customisable with a selection various inputs being possible for the decision process. The framework has been tested, using a case study to dynamically control arrivals for a series queuing systems, to demonstrate its ease of use and flexibility. Proposed in [95] is a fuzzy logic learning algorithm for
Chapter 3 Network Management Architecture

predicting real time cell loss probability (CLP) for self-similar traffic so that satisfactory CAC can be achieved. The necessary training data for the algorithm is generated using mathematical expressions derived for a self-similar ATM traffic model. The results presented show that the predicted CLP closely matches the theoretical CLP and that the buffer size required can be predicted for a known CLP.

Works such as [84, 85, 86, 87] address network congestion control and avoidance. Gan et al [84] propose a genetic algorithm and fuzzy logic tool for congestion avoidance in high-speed networks. To demonstrate the effectiveness of the design tool and improved congestion avoidance performance it was used to improve random early detection (RED) queue management mechanism for TCP/IP networks. Zhang and Ma [85] introduce a drop based congestion control mechanism using fuzzy logic for differentiated services networks with packets being dropped based on service precedence. Likewise in [86] Chrysostomou et al propose a fuzzy explicit marking (FEM) queue management scheme for congestion control for differentiated services networks. The authors verify through simulation the effectiveness, robustness and scalability of the proposed scheme when compared against a RED queue management scheme with and without the implementation of explicit congestion notification (ECN). Congestion control for ATM networks is investigated in [87] through the use of a fuzzy based CAC scheme. The proposed fuzzy CAC method is compared against traditional CAC schema, which employs capacity estimation or buffer thresholds. Simulation results shown that the fuzzy based CAC achieves better congestion control than conventional CAC methods while maintaining the system QoS requirements.

Pithani and Seth [88] have applied fuzzy logic to distributed dynamic routing algorithms. They have used fuzzy sets to represent uncertainty with regards to network status over time and these uncertainty estimates are used to select minimum delay paths for routing. The use of fuzzy set delay for route selection has been applied using the Bellman-Ford algorithm. Simulation results have shown the effectiveness of the fuzzy delay based routing in terms of routing algorithm delay performance with varying network loads. Aboelela and Douligeris [89] have assigned a multimetric fuzzy link cost to weight B-ISDN links for route selection. Simulation results express the efficacy of this multimetric route selection scheme as network utilisation and throughput are increased.
Chapter 3 Network Management Architecture

when compared against a crisp routing scheme [96]. Wong and Wong [90] have proposed a fuzzy decision based protocol as an extension to DSR protocol for routing over ad hoc networks. Route selection is made using a fuzzy controller that takes node speed, loss and end-to-end delay as fuzzy inputs to determine the fitness of a route. Traffic is assigned to route based on its fitness value with higher priority traffic preferring the ‘most fit’ route. Simulation results exhibit an improved performance with regards to delay for the fuzzy based DSR when compared against basic DSR.

Pasupuleti et al [91] propose an adaptive routing algorithm that dynamically assigns link costs using fuzzy logic. Traffic over the network is re-directed towards less congested nodes. Link costs are assigned based on the current network congestion status and a node’s present queue size. The performance of the fuzzy based routing was compared against individual route selection metrics and the fuzzy scheme outperformed the single metric selection in terms of throughput and delay for varying load conditions and topologies. Alandjani and Johnson [92] have presented a fuzzy based routing scheme for ad hoc networks that considers traffic priority, network state and resource allocation as fuzzy input metrics. The route discovery process generates a set of maximally disjoint paths with fuzzy logic being used to determine how to distribute traffic over the routes. Simulation results have shown, when compared against other routing protocols such as DSR, that higher reliability and lower delay is attained for high priority traffic and throughput is increased for all traffic levels.

The successful application of fuzzy logic for managing communications networks has been used as a motivation for developing the proposed fuzzy logic based management architecture presented in this thesis. CAC schemes are used for controlling what traffic gains access to a network and are responsible for limiting the number of sessions allowed. CAC methods determine whether a new call can be serviced without degrading the performance of existing sessions. It is the responsibility of the CAC method to be aware of the resource requirements to needed support existing and new sessions and to do this the CAC method must be aware of the current network performance. Fuzzy logic has been effectively used to implement CAC schemes for network management, as previously discussed in works such as [82] and rely on multimetric network observations for fuzzy decision based access schemes. This
methodology is adopted in developing a fuzzy logic based broadcast limiting algorithm that restricts the flood of route discovery packets to stable routes, using instantaneous network conditions in the fuzzy decision process. Using both channel and self awareness network nodes decide using fuzzy logic to selectively drop route request packets, as in [76] where a percentage of incoming ATM cells are selectively dropped, on the premise of promoting stable route generation. This can be likened to the action of a CAC as the broadcast limiting algorithm controls what routes are generated for data forwarding and thus is responsible for what routes are made available for data sessions.

A routing protocol must be adaptive and to react to network changes its must interpret present network conditions with nodes making distributed local routing decisions. For MANETs fuzzy logic has been used to implement adaptive routing schemes using either individual [88] or multiple metrics [89, 90, 91] to assign fuzzy link costs that are used for route selection. The simulation results presented for these schemes shows the effectiveness of fuzzy logic in assessing link quality. This provides the motivation for using fuzzy logic to adaptively select links to include in the route discovery process. The common metrics used in the fuzzy evaluation of link quality are traffic load and delay, where delay is either queuing delay or end-to-end delay. These metrics are correlated and give a similar interpretation of the network condition. If a network load is high then large queuing delays are likely to be experienced, which will also increase end-to-end delays. Conversely, if the network load is low then the associated queuing and end-to-end delays will also be low.

In order to extract a more complete interpretation of a node's state and that of its local neighbourhood the metrics link SNR, path length (number of hops), node energy and node load are used. The node energy is indicative of node lifetime and nodes with small lifetimes should be avoided, as routing over such nodes will only serve to drain battery life at a faster rate, which will lead to link breakages when the node dies. The node load is used to signify how busy a node is, with a high load inferring that the node is either generating or forwarding large volumes of traffic or that the local channel is busy, consequently such nodes should not be included in route generation. The link SNR indicates the received signal strength and large SNR values would indicate strong connectivity and thus be desirable for routing. The path length metric is used to avoid
Chapter 3 Network Management Architecture

routing over excessively long paths as the greater the number of nodes in a path the more likely it is to have link breakages due to node mobility.

3.3 Network Management using Policy

With the ever-increasing size, complexity and move towards seamless heterogeneous networks there is a need for an automated network management scheme. The use of policy based networking for network management and control has been researched extensively and employed in diverse domains such as: QoS [99, 100, 101], security [102], admission control and network management in optical networks [103], mobility management and access control in heterogeneous networks [104, 105] and routing [97, 98].

3.3.1 Policy Based Networking as an Approach to Network Management

Ponnappan et al [99] have presented the design, implementation and performance evaluation of a policy based QoS management system for the IntServ/DiffServ based internet. The proposed policy management scheme is aimed at controlling different types of policies and the sharing of policy information with a view to providing end-to-end QoS. Simulation scenarios examine the policy server response times for varying load conditions and have identified scalability issues that must be addressed while proposing an appropriate policy caching mechanism as a solution. Bohm and Braun [100] detail on the application of policy based mechanisms for QoS control in the IP multimedia subsystem (IMS) of 3rd generation UMTS network and show how current frameworks can be extended to provide end-to-end IP QoS for the IMS. In [101] the performance limitations of the IETF recommended policy based management system are investigated. As an alternative the authors propose a new multi-tiered policy based management framework with simulation results indicating that this approach yields a high performance and scalable policy based management framework.

Presented in [102] is a policy based access control framework for large networks. They propose three levels of access control that manages a large number of filtering rules
Chapter 3 Network Management Architecture

in multiple firewalls so that specific security requirements can be satisfied. The authors discuss the associated design and implementation issues for the policy based access control security aware framework. A policy based management framework for service provisioning in optical network is proposed in [103] using service level agreements. The authors describe an overall policy based framework necessary for managing the network with customers being allotted a unique identifier once a service contract has been established with the network operator, with this identifier being used in the policy management scheme to provision user service.

Murray et al [104, 105] have demonstrated how policy based management using capacity surfaces can be used for access control and inter-system mobility management in a heterogeneous wireless network. The proposed scheme is evaluated for EDGE and UMTS technologies and compared against a random allocation scheme with simulation results demonstrating a load balancing effect and better QoS levels with a policy based management scheme.

Policy based adaptive routing is addressed in [97] to improve the self managing capabilities and routing protocol adaptability to dynamic topologies and network conditions. A policy to switch between a the proactive protocol DSDV and the reactive protocol DSR is implemented using application latency sensitivity, loss-tolerance and average nodal speed as policy parameters in the protocol selection decision. An adaptive route cache entry timeout policy is presented based on the activity of the nodes at either end of cache path entry. This adaptive timeout is based on node position (assuming nodes have GPS capability), the node direction and nodal speed with these values being updated periodically. Rather than using the shortest number of hops as the route selection metric it is proposed to use a minimum energy path cost. This path cost is based on the required energy to transmit on each radio link in the complete source-destination path. The simulation results presented that use the minimum path energy as the route selection criteria exhibit reduced power consumption however, the throughput is slightly lower than a network that uses the shortest number of hops as the route selection metric. The concept of dynamic updating of aggregated routing information for hierarchical QoS routing in ATM networks is investigated in [98]. The authors state that a time-based update interval for aggregating route information is not adequate for achieving QoS when
Chapter 3 Network Management Architecture

faced with dynamic network traffic. The proposed dynamic update policy is compared against: a time based update approach, a full update policy that aggregates and distributes information when any of the QoS parameters change value and a logarithm of residual bandwidth update policy that aggregates and distributes routing information based on the logarithm of residual bandwidth of a link. The simulation results demonstrate the effectiveness of this dynamic update policy when compared against the other update schemes in reducing the overhead associated with the aggregation and distribution of hierarchical routing information.

Policy based network management is based on defining a set of global rules that defines how a network must operate. Network managers must observe the health of their network through the use of monitoring [75] and distribute policies to effectively manage the network. This can be said to be analogous to MANET nodes monitoring their own state and their local channel and dynamically modifying their routing behaviour to adapt to current network conditions. This equivalence has served as the motivation for applying the principles of policy based management to a MANET environments. Policy based networking is implemented via a centralised entity that manages network monitoring via feedback from network elements. A MANET is a distributed system and has no knowledge of the global network state, network nodes are aware of their local state only and cannot make decisions regarding the global state. There is no temporal relation for event occurrence in a distributed network, unlike a centrally controlled network where events are ordered as a consequence of their chronological occurrence [97]. Consequently, for a distributed system such as a MANET policy based management must be implemented locally. In this thesis policy based management has been used to implement a node congestion control mechanism that adaptively alleviates network congestion.

3.4 Proposed Network Management Architecture for MANET Nodes

Presented in this section is the proposed adaptive network architecture that is used to implement dynamic route management strategies for MANET nodes, which aims to furnish nodes with an adaptive intelligence that augments node routing functionality. In
Chapter 3 Network Management Architecture

wireless networks, nodes are competing for access to a limited radio channel that is shared by all nodes in the network communication space. Unnecessary or uncoordinated transmission leads to a wasting of already restricted resources and interference, which leads to collisions that reduce network throughput. To address these problems nodes must have the ability to adaptively manage their behaviour. As an ad hoc network functions in a decentralised manner, nodes must be capable of adaptive behaviour as there exists no omnipresent network controller that can instruct node coordination and routing functionality as a consequence of network wide observations. Providing MANET nodes with the capability to observe both themselves and their external environment in conjunction with the capacity to derive/execute responses based on these monitorings negates the need for an external centralised network controller.

To achieve adaptability nodes must have an information infrastructure that generates system knowledge, consisting of both self and local channel awareness. Using this system knowledge routing decisions can be inferred. The decisions taken can be used to modify a node’s routing capability and routing protocol parameters. Shown in Figure 3.1 is the proposed novel network management overlay architecture that MANET nodes will implement to extract self and local neighbourhood awareness. This architecture comprises of the following entities:

- Congestion State Monitor – nodes monitor their congestion status and if they are approaching congestion or are congested then action is taken to alleviate it
- Route Flood Limiter – this is a fuzzy logic system that is incorporated with the route discovery mechanism to restrict network broadcast floods.
- Cache Entry Weight Assignment - The fuzzy system is used to instruct caching decisions and weigh routes for path differentiation, using a fuzzy utility function
- Cache Entry Expiration Policy – this is an adaptive cache entry timeout policy that uses the link characteristics to dynamically calculate a timeout value when a route record is cached
- Merging Node Awareness – for integrating merging network node with an existing network

The following sections describe in detail the proposed network management architecture and the entities used to implement it.
Chapter 3 Network Management Architecture

Figure 3.1 Network Management Architecture For MANET Nodes
Chapter 3 Network Management Architecture

3.4.1 Congestion State Monitoring using Policy

Policy based networking and fuzzy logic operate in a similar manner with both techniques being used to make a decision based on a set of input conditions. The principles of policy based networking and fuzzy logic could have been used interchangeably in the architecture developed and presented in this thesis. Section 3.2 discussed the use of fuzzy logic based approaches towards network management, where fuzzy logic was used at network level for global decision making and at individual node level for local decision making. Policy based networking is traditionally used at a global level for network management, where a centralised entity determines and invokes the policy decision. To demonstrate that a policy based architecture can, like fuzzy logic, be effectively implemented at a local level it was decided to implement the congestion state monitoring method using policy.

A routing protocol must balance the network load to achieve fairness in resource usage and maximise throughput. Current methods for load balancing take a proactive approach that involves constructing paths over lightly loaded nodes in the route discovery phase of the protocol [41, 47, 48]. The paths discovered are then used for data traffic routing by the network. Consequently, these paths can become congested and contribute to a degradation in network performance. The Congestion Monitor approach proposes a reactive method using a distributed policy approach that enables nodes to periodically determine their congestion status. As a node approaches congestion or determines itself to be congested, it takes action to alleviate this condition by making its neighbours aware of its congestion state thereby causing these neighbours to route traffic over paths that do not involve this congested node. This response to the congestion notification will lessen the node's congestion status. The aim of this adaptive policy is to dynamically redistribute network load away from congested nodes and consequently improve queue and transmission performance, while reducing transmission queue latencies and thus optimising routing protocol performance.

3.4.1.1 Policy Engine

The proposed policy system architecture, located on individual nodes, for congestion state monitoring is shown in Figure 3.2. Policy management is typically
Chapter 3 Network Management Architecture

implemented in a centralised manner however, the policy method presented here is realised using a distributed approach to suit the distributed operation of an ad hoc network. The approach acts on the congestion level of a node as this affects its local neighbourhood only. Consequently, both the PEP and PDP, i.e. the policy engine, are implemented on individual nodes. The data required by the congestion monitor is maintained in a policy repository. The policy repository, as per Figure 3.3, evaluates the Channel, Queue and Transmission states that are used by the policy rules. The policy engine is run at $t$-second intervals by every node present in the network to determine the current node congestion status and the congestion control algorithm is run if necessary.

The policy repository makes available these policy parameters to the PDP in the congestion state determination process. It is the PDP that resolves the nodes congestion level. This level is set to one of the possible output states as shown in Figure 3.3. The PEP is informed of the congestion level and if this state is set to HIGH a congestion notification ($CN$) is sent to the nodes local neighbourhood or if the state is set to MED-HIGH a congestion warning ($CW$) is sent, otherwise no action is taken by the PEP. The neighbourhood responds accordingly thereby alleviating the nodes congestion state to an acceptable level. The policy rules used in the congestion monitor define the congestion state that is to be set in response to a combination of input state values.

![Figure 3.2 Congestion Monitor Policy Engine](image-url)
Chapter 3 Network Management Architecture

Map Measurements to State Levels

<table>
<thead>
<tr>
<th>State Levels</th>
<th>Map State Level to Repository Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW</td>
<td>Channel Queue</td>
</tr>
<tr>
<td>MED</td>
<td>Queue Transmission</td>
</tr>
<tr>
<td>HIGH</td>
<td>Policy Repository</td>
</tr>
</tbody>
</table>

**Figure 3.3** Policy Repository State Evaluations

### 3.4.1.2 Policy Repository State Interpretation

The channel state, $CS$ as per equation 3.1, is the percentage time that the node is busy in a $t$-second interval. A node is said to be busy if it is contending for channel access, transmitting or receiving (either receiving directly or overhearing).

$$CS = \frac{\sum c_t + tx_t + rx_t}{t} \times 100$$  \hspace{1cm} 3.1

Where: $c_t$ is total contention time, $tx_t$ is total packet transmission time and $rx_t$ is total packet receive/overhear time in a $t$-second interval.

The queue state, $QS$ as per equation 3.2, is a ratio of the number of packets that were presented at the network interface queue but were dropped, as the queue was full, to the number of packets that were actually placed in the queue.

$$QS = \frac{\# q_{\text{off}}}{\# q_{\text{in}}}$$  \hspace{1cm} 3.2

Where: $\# q_{\text{off}}$ is the number of dropped packets and $\# q_{\text{in}}$ is the number of packets placed on the queue, in a $t$-second interval. This measure gives an indication of the performance
of the queue. If this ratio is high then the queue can be viewed as having performed poorly as it is dropping a large number of packets compared to the number it is placing in the queue. This can be construed to mean that the queue was congested (full to capacity) for a large period of time over the previous interval. This indicates that the node’s local channel is busy and/or this node itself is generating a large volume of traffic. If this ratio is low then the node has not approached capacity level or the local channel is relatively quiet so channel access times are small. Using this measure of queue performance is more indicative than simply taking the instantaneous queue value as is done in [41, 47, 48]. If the current queue size value is close to capacity then the node is considered congested. However, this could simply be that the node is simply trying to gain channel access and when it achieves this it could clear its queue in a matter of a few hundred microseconds, assuming IEEE 802.11b is being used, consequently the node should be seen as being in a non-congested state. The queue state ratio looks at the queue packet acceptance rate over a period of time and gives a clearer measure of how the queue has performed whereas taking the present queue size can give a false impression of queue functioning.

The transmission state, \( TS \) as per equation 3.3, is a ratio of the number of retransmissions to successful transmissions (not including broadcast packets).

\[
TS = \frac{\#rtx}{\#s_{tx}}
\]

3.3

Where \( \#rtx \) is the number of retransmissions and \( \#s_{tx} \) is the number of successful transmissions in a \( t \)-second interval. A high transmission state ratio implies a large number of packets are retransmitted, which points to a large number of collisions on the channel and indicates that this area of the network has at present high levels of traffic passing through it – i.e. it is congested. In contrast a low transmission state ratio implies that the node’s channel is quiet. These states are mapped to LOW, MEDIUM or HIGH levels in the policy repository, see Figure 3.3, and are used for policy rule evaluation by the PDP, where a congestion level for a node, which incorporates the node’s performance and its perception of the channel around it, is determined.
3.4.1.3 Local Neighbourhood Reaction to Policy CW/CN Messages

When a node assesses itself as approaching congestion (congestion level = MED-HIGH) or is congested (congestion level = HIGH) it must make its local neighbourhood aware of this by sending an appropriate congestion message. If a node is in the MED-HIGH congestion state it sends a CW message and for a HIGH state it sends a CN message as a broadcast to its local neighbours.

The neighbourhood responds to CW broadcast by examining its network interface queue and its cache. If the neighbour node has a cache entry that can be used to divert traffic away from the congested node it modifies the headers of the packets concerned in its network interface queue and sends them via other network nodes. However, if no such cache entries are available then the node continues to use its present route paths. When a neighbour node receives a CN message (congestion level = HIGH), and is unaware of alternate routes not involving the congested node it removes packets from its network interface queue that are to be routed over this node and removes all cache entries concerning the congested node. Otherwise the packets are rerouted as in the congestion warning response.

The purpose of this adaptive packet rerouting/dropping is to reduce the congestion level of the effected nodes and to distribute traffic over lightly loaded nodes so that the performance of individual network nodes may be optimised in terms of load balancing. With congestion monitoring the average end-to-end network latency is improved as congested network zones are identified and traffic is possibly rerouted around these areas, giving congested nodes an opportunity to recover their status to a non-congested state. This allows data traffic to be routed through the network nodes interface queues with reduced delays.

3.4.2 Route Flood Limiter

Routing is an essential function for multihop communications management in ad hoc networks. Table-based routing algorithms used in proactive protocols are computationally intensive and require periodic transmission of status information
Chapter 3 Network Management Architecture

amongst all network nodes. Also, routing table volume swells with network size and can be large for dense networks. Therefore, to cope with possibly rapidly changing topology, reactive protocols, such as DSR or AODV, generate routes for unknown destination paths on an as needs be basis. So as to avoid the continual demand for route discovery in reactive protocols, network nodes record previously determined routes in a route cache. DSR, as an example, stores the complete source-destination path and can have multiple entries for a single destination. A route discovery attempt can possibly result in several paths being uncovered for a single destination. As nodes often have a finite capacity path cache, it may not be possible to store all paths. So as to influence productive caching decisions a fuzzy logic system is applied to the route discovery technique to curb non-optimal network floods. This action causes a cessation in the generation of low quality routes as only paths with good routing metrics are selected for the rebroadcast of route discovery packets. Consequently, route query packets arriving at the necessary destination node, or at some intermediate node with knowledge of the destination node, generate high quality route replies. The purpose of the fuzzy logic based decision algorithm is to weigh an individual link as a path to the necessary destination is being constructed. If this link is deemed suitable by the fuzzy logic system it is added to the path and route construction continues. The fuzzy controller is used to instruct caching decisions and to optimise route selection as only good quality links are recorded in source/destination paths.

DSR has been used in this thesis to illustrate the effect of multimetric route request limiting decisions using fuzzy logic. Results of the performance evaluation are presented in chapter 5. A node initiates route discovery when it wishes to transmit a packet to a destination that it does not have a cache record for. To do so, the node broadcasts to its local neighbourhood a route request for the necessary destination. Any node receiving such a packet adds its own address to the route record and rebroadcasts the packet to its neighbour zone if it has not already processed this packet or its address is not already listed in the route record. This route request packet will arrive at the required destination or at some intermediate node that has knowledge of a path to that destination whereupon a route reply is returned to the route request initiator node that lists the path from source to destination, which is then cached by the source node, as shown in Figure
Chapter 3 Network Management Architecture

3.4. A single route discovery can result in possibly multiple routes for a destination. However, the flooding nature of a route request can harshly influence the performance of a network, as several nodes may initiate floods concurrently, generating bursts of broadcast traffic through the network resulting in collisions and retransmission attempts that can cause congestion in a network. By implementing effective caching and route request limiting the need for route discovery can be minimized and the performance of the network improved. Each network node is assumed to have a fuzzy engine, as per Figure A.1, with a corresponding rule base that has been manually tuned for optimal performance based on extensive simulation evaluation for varying network sizes, topologies, node densities, mobility patterns and traffic flows. The fuzzy rule base was optimised through the modification of both rule antecedents and consequents, while also varying the membership functions cross points. The presented fuzzy system is tuned to provide optimal performance for the network scenarios under test with a common rule base and membership function set being used for all test scenarios. To universally tune the fuzzy system and make it applicable to general network scenarios, adaptive learning techniques should be used to dynamically tune the fuzzy rule set and membership function cross points. Such an approach would lead an optimal set of rules and membership functions for individual scenarios.

Figure 3.4 Route Discovery/Reply Process
3.4.2.1 Fuzzy Routing – as applied to DSR

In DSR, as a route request is flooded through the network, nodes append their own address to the route record and rebroadcast the request. It is proposed that nodes, which appear in this route record should determine whether or not to continue with the route discovery process, as shown in Figure 3.5.

![Figure 3.5 Fuzzy Route Request Limiting And Caching](image-url)
Route metrics that are used to make this decision are link strength, energy available at a link vertex, current network interface queue size, and number of hops currently in a path. It is proposed here that combining several metrics into a single decision makes a routing protocol more robust and introduces an element of self management as nodes are aware of their current state, based on remaining battery life and network interface queue size and make routing decisions based on this awareness. This self awareness is combined with the received SNR link value and the present path length so that an informed routing decision, as regards continuing to flood the route request, can be deduced, with the aim of optimising routing performance. The decision to continue with a network broadcast will be determined via a fuzzy logic system with the caching parameters being applied to a fuzzifier that translates them into fuzzy sets. The fuzzy sets are used to appraise each constraint as being Low, Medium or High, assigning each a value between [0,1], using triangular membership functions of the form shown in Figure 3.6. These evaluations are passed to a fuzzy inference engine that applies a set of fuzzy rules that determines if a route is apt for caching or not. If a route is deemed suitable then the route request is rebroadcast and the node extracts and caches the route record. When a route request arrives at the necessary destination a route reply is generated and sent to the initiator of the route request by reversing the path stored in the route record.

![Figure 3.6 Fuzzy Triangular Membership Function](image)
Chapter 3 Network Management Architecture

The broadcast flood limiting algorithm decides using fuzzy logic whether to continue the rebroadcasting of a route discovery flood. The fuzzy logic rule base has been designed so that the rule consequents are either terminate broadcast (TB) or continue broadcast (CB). This can be simplified to yes/no decisions.

- **TB** – do not store route information as link is considered to be unstable, terminate route request flood
- **CB** – store route information and continue with broadcast flood if node is not the necessary destination of the route request packet

Example fuzzy rules for both TB and CB conclusions are shown in Table 3.1.

<table>
<thead>
<tr>
<th>Table 3-1 Sample Fuzzy Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF (LS=HIGH) AND (NE=HIGH) AND (NH=LOW) AND (NQ=LOW) THEN CACHE</td>
</tr>
<tr>
<td>IF (LS=LOW) AND (NE=LOW) AND (NH=HIGH) AND (NQ=HIGH) THEN NO CACHE</td>
</tr>
</tbody>
</table>

Where, 
- LS = Link Strength
- NE = Node Energy
- NH = Number of Hops
- NQ = Current Network Interface Queue Size

The fuzzy rule set is evaluated using the min-max technique [73] to calculate numerical values representing the truth for each consequent, as this is suitable for evaluating fuzzy decisions that are equivalent to yes/no decisions. With the min-max technique the degree of truth is determined for each fuzzy rule and the minimum fuzzy metric value is selected as the rule truth strength. To determine whether to continue or terminate a broadcast the rule base is divided into TB and CB subsets with the maximum truth value of each subset being selected as the overall rule strength for the subset. These strengths are then compared and the largest value is selected as the decision.

To summarise: as a consequence of the broadcast nature of route discovery techniques in reactive routing protocols such as DSR, network resources can be unnecessarily used in this network wide propagation that often leads to the selection of unstable paths. Unstable path are classified as paths that have a large associated signal loss, consists of low-energy nodes, heavily loaded network interface queues and a high number of hops or paths spread over a large distance between source and destination. So
as to eliminate unsuitable paths from the route discovery/reply process and to optimise
caching decisions broadcast floods are only continued if a node’s fuzzy system indicates
that it is valid to do so, as per the algorithm shown in Figure 3.5. Likewise a node caches
route information extracted from packet headers that are received over good links.

3.4.3 Cache Entry Weight Assignment

Caching is an important part of reactive MANET routing protocols. In an ad hoc
network nodes must cooperate to dynamically establish and maintain fresh routing
information and forward packets for each other to allow communication between nodes
not directly within wireless transmission range. Rather than using the periodic exchange
of routing information as in proactive routing, reactive routing protocols initiate route
discovery to find a route to some destination node only when a sending node originates a
data packet addressed to that node. To avoid the need for performing route discovery
before each data packet is sent, reactive routing protocols cache previously discovered
routes. However, this caching behaviour introduces the need for strategies to manage the
contents of the node cache as network nodes move in and out of wireless transmission
range of one another, possibly leading to link breakages and thus invalidating some
cached routing information. For a routing protocol to function effectively it should select
the best route over which to transmit data packets. Therefore when caching routes, be it
next hop information or entire paths, routes should be weighted to indicate their quality.
When multiple paths are available for a single destination the protocol must be able to
decide which path is best. In DSR and AODV for example, the path with the least
number of hops is selected as the best path however, this metric is not indicative of the
quality of the path as another route with an equal number of hops or a path with more
may consist of higher energy nodes, better link SNR or queue levels. The fuzzy $CB$
decision can be used to assign a weight to a path. Three methods are proposed as possible
ways of assigning path weights, using the fuzzy metrics - current number of hops ($H$),
link SNR ($S$), node battery energy ($B$) and present network interface queue size ($Q$):

- Use the value of the $CB$ decision as a weighting – *Fuzzy Value*
Chapter 3 Network Management Architecture

- Use the difference between the maximum rule strengths for the TB and CB rule subsets - *Fuzzy Difference*
- Use the strength difference as above but incorporate it in a utility function that includes the metrics used to make the fuzzy decision – *Fuzzy Utility Function.*

The utility function is defined as:

\[ \text{FUF} = \alpha DS + \beta DB + \sigma DQ + \delta DH \]  \hspace{1cm} (3.4)

Where \( \alpha, \beta, \sigma, \delta \) are scaling factors representing the relative importance of their associated fuzzy metric (these are assigned the values 0.6, 0.2, 0.1, 0.1 respectively) and \( D \) is the strength difference between the maximum rule strengths for the TB and CB rule subsets. The weight values assigned to the fuzzy metrics were derived based upon simulation observations. From extensive simulation it was evident that the signal strength measurement had the strongest effect on link lifetime. A high signal strength measurement can be taken to indicate that nodes are moving towards each other and/or there is little destructive obstacle interference on the communications path. A signal strength measure that just exceeds the sensing threshold reflects a poor link and/or that terminating nodes are moving away from each other. Accordingly, the signal measurement was afforded the largest weighting factor in the fuzzy utility function. The battery energy metric was observed to have a lesser effect as node operations, such as transmissions and receptions, deplete the battery relatively slowly. The metrics queue size and hop count were assigned the least weight values as nodes can clear their queues by transmitting packets in the order of microseconds when channel access is achieved. To route packets over paths with a large number of hops it is essential that the intermediate links have high signal strength measures and that the nodes have reasonable energy values otherwise it is unreasonable to route over such paths as links breakages are more likely due to node movement and battery exhaustion. A fuzzy logic defuzzification process aggregates the rule consequent strengths for the entire rule base, however the proposed cache entry weighting policy is only invoked after a CB decision has been made so in reflecting the quality of a path only the relative or absolute strength value of the CB decision is considered. Such a cache entry weighting scheme allows a node to select the best possible source-destination path from multiple entries for a single destination based on a combined comprehensive metric rather than a single metric such as 'select the
Chapter 3 Network Management Architecture

shortest path'. The inclusion of this collective weighting metric as the path selection criterion configures the routing protocol to adaptively select the best path based on current network conditions.

3.4.4 Cache Entry Expiration Policy

For reactive routing protocols aggressive caching of route information can considerably improve network performance. However, preventing route cache staleness is a serious problem for such protocols. A cache record may become invalid due to topological changes such as nodes moving out of wireless transmission range of each other, a node is not notified when a cache entries becomes invalid, unless the node attempts to use the entry for routing a packet. Storage of any routing information can only persist in a node's cache for a finite amount of time before it must be expired. For a static cache entry timeout policy, nodes simply disregard cache entries after a set time whereas an adaptive timeout policy uses the link characteristics to dynamically determine a timeout value when the cache entry is created. However, if the timeout value is not well tuned, network performance can degrade as cache entries are removed either prematurely or too late from the cache. For the work presented here, cache entries are weighted during the route discovery process so the timeout policy can be combined with the fuzzy utility function weight. This does not require the nodes to store any additional overhead nor does it require the inclusion of previous lifetime value. DSR specifies a static timeout value that remains constant for the network duration. Such a value may possibly be excessively long and thus cause the persistence of stale cache information or on the converse it can be too short thereby promoting the premature expulsion of accurate routing information. An adaptive policy, such as the one presented here, aims at dynamically adapting the routing protocol cache entry expiry parameter with expiry times being generated as a consequence of path quality. Taking the static timeout value (STV) the adaptive policy modifies it using the weight determined by the fuzzy utility function, as discussed in the previous section, according to the following policy:

\[
\text{If Fuzzy Weight} \geq 0.25 \text{ Then } \text{Timeout} = \text{STV} + (\text{STV} \times \text{Fuzzy Weight}) \\
\text{If Fuzzy Weight} < 0.25 \text{ Then } \text{Timeout} = \text{STV} - (\text{STV} \times \text{Fuzzy Weight})
\]
The timeout value is determined by increasing or decreasing the $STV$. A fuzzy weight threshold value is necessary to decide whether to enhance or lessen the static timeout. The threshold value of 0.25 is a manually tuned simulation parameter that was optimised through extensive simulation runs. When a cache entry is updated the timeout value should be recalculated using the new fuzzy metrics. The $STV$ is either increased or decreased depending on the path quality. The purpose of this is to improve caching performance by retaining current routing information for as long as possible while expiring cache contents that are likely to become obsolete over a short period of time.

### 3.4.5 Merging Node Awareness using Fuzzy Logic

Both proactive and reactive routing approaches can conceivably suffer from excessive generation of control traffic, in particular in the flooding based route discovery process and periodic route table dumps. Merging network nodes can indiscriminately join a network and will initially need to construct a routing table/cache before they can begin transmitting self-generated data traffic. During this start-up process a merging node can contribute greatly to the overall control traffic overhead. Merging nodes should garner valid routing information from their local neighbourhood by procuring copies of their route tables rather than using blind flooding. However, consideration must be given to the impact of the cache provisioning scheme on the nodes that are required to provide copies of their routing tables. Such nodes are encouraging data traffic to be routed over them, which can lead to or worsen the congestion state of a network.

The proposed merging node awareness policy is a scheme that allows existing network nodes to decide, using fuzzy logic, whether or not to make available its route cache to merging nodes. If a node is currently congested, has low battery power and has comparatively small number of one hop neighbours then it is not in the best interests of such a node to encourage merging nodes to route over it whereas a node that is not congested, has high battery energy and a relatively large local neighbourhood would decide to provide a merging node with a copy of its cache. With existing nodes making a decision whether or not to impart their caches the impact of merging nodes is distributed over nodes that are suitable as next hop neighbours and will avoid contributing to already
congested network areas, burdening low energy nodes with additional traffic and nodes that are actively routing for a large local neighbourhood.

The architecture of the fuzzy logic controller used to decide whether an existing network node should provide a copy of its route cache is based on that shown in Figure A.1. The fuzzy inputs are current congestion level, neighbour reachability ratio and remaining battery energy. A node's congestion level is periodically determined, as described in section 3.4.1 and is a node's perception of its own performance and that of the channel around it. A node's neighbour reachability is the ratio of the number of one-hop neighbours to the number of neighbours that can be reached indirectly via the one-hop neighbour set. If this ratio is low then the node has a small one-hop neighbourhood and is relying on relatively few nodes to route its traffic. Such a node could experience large channel access times, collisions and retransmissions if its local neighbourhood is generating large volumes of data traffic and is dependent on this node for forward routing. The neighbour reachability ratio is determined from a node's current cache contents where a one-hop set and a reachability set are extracted. These crisp input metrics along with remaining battery energy are used as fuzzy inputs that are fuzzified using triangular membership function. The fuzzy rule set is split into a YES and NO subsets with the min-max method being used to determine either a YES or NO decision as the fuzzy output. A YES decision indicates that a node should transmit a copy of its cache and a NO decision prevents a node from transmitting a copy of its cache. When nodes transmit their cache contents they also include a measure of the strength of fuzzy decision (the difference between the fuzzy YES/NO rule evaluations). This allows merging nodes to select the best possible subset of existing cache entries from several nodes.

3.5 Suitability of Proposed Management Architecture as a General Approach to Route Management in MANETs

The proposed management architecture is applied to DSR, which is a unicast topology based reactive routing protocol, for the purposes of evaluation with system performance results and analysis being presented in chapter 5. However, this
management architecture can be extended for application to other routing protocols and this application is described based on Figure 2.1, which shows a genealogy of some of the most common unicast and multicast MANET routing protocols in existence.

The congestion state monitoring method can be utilised directly with both unicast and multicast MANET routing protocols. This congestion control method requires nodes to reroute traffic away from congested network zones and to remove route record entries to prevent subsequent data from being transmitted over congested nodes. As both multicast and unicast protocols rely on the use of route tables, caches or trees to maintain fresh routing information the removal of route records from these structures prevents the congested node from being used for routing.

The merging node awareness scheme is suitable for direct application to unicast topology based proactive, reactive and hybrid routing protocols. Proactive routing protocols rely on periodic or differential topology updates to maintain neighbour connectivity, reactive protocols generate routes on demand and hybrid based routing employ both proactive and reactive routing techniques for topology awareness. As topology based routing protocols utilise route tables or route caches to store fresh routing information existing network nodes can use the merging node awareness algorithm to furnish merging nodes, which have no topology knowledge, with route information. Although proactive routing protocols provide routing information via route record dumps, the merging node awareness algorithm will provide new network nodes with the best available subset of existing one hop neighbours over which to initially route packets. For geographical based routing protocols the merging node awareness algorithm can be extended to include location information in the fuzzy decision so that the nodes which provide routing information are the geographically closest subset of available neighbours. With multicast routing protocols the merging node awareness algorithm can be limited to multicast group leaders so that merging nodes construct route information via these multicast sources.

The fuzzy logic based broadcast flood limiting algorithm, which promotes stable route generation, is appropriate for use with unicast topology based reactive routing protocols and reactive multicast routing protocols as such protocols rely on broadcast floods for route discovery. Reactive routing based protocols that record multiple paths for
Chapter 3 Network Management Architecture

a single destination, such as TORA, can employ the cache entry weighting policy for path differentiation and can use the cache entry expiration policy to purge stale routing information. For reactive routing protocols that record a single entry for a destination, such as AODV or MAODV, the cache weighting policy can be used in conjunction with the sequence number to indicate the route quality. If a source node receives multiple route replies with the same sequence number then the path weights can be used to differentiate between the routes allowing the source node to record the path that has the most current sequence number and is the best of all the available paths at present. For proactive routing protocols, such as DSDV or OLSR, route discovery floods are not used however, the flood limiting algorithm could be used to assign link weights, so that when topology updates are broadcast nodes are also aware of the associated link quality. For geographical based routing protocols that use broadcast floods in the route discovery process, such as GRA, the flood limiting algorithm can be extended to include location information so that routes are generated over a set of stable nodes which are geographically closest to the source node.

3.6 Management Architecture Limitations

The architecture presented in this thesis is not billed as a solution that will be applicable to all network topologies, environments and operating conditions. However, it is a general approach to route management for MANETs, which can be effectively used with existing routing protocols over a broad range of scenarios for improved network performance. The management methods developed when incorporated with existing routing protocols, such as DSR, does lead to improved protocol performance. A range of environment scenarios have been simulated and tested, with the proposed dynamic route management strategies being incorporated with DSR for evaluation purposes. However, these management strategy operating parameters have been optimised for performance over these scenarios and therefore cannot be considered as generic parameters.

The congestion state monitoring approach has been developed to assuage localised network congestion. This method looks towards rerouting traffic away from congested network zones through the use of broadcast congestion warning and
Chapter 3 Network Management Architecture

notification messages. Such a scheme is ideally suited for large dense networks as more nodes are available over a larger area, thus making it possible to redistribute traffic over non-congested network zones as additional routes exist for load redistribution. For small scale densely populated networks this method would suffer. With such networks, nodes would be in close proximity to each other causing increased interference and large channel access times. While nodes outside the localised congested zone may have routes available for traffic redistribution they can possibly be prohibited from participating due to severe operating conditions. Likewise, with sparsely populated networks there may be few alternate routes available that do not include the congested nodes, making it difficult if not impossible to reroute traffic. Employing congestion state monitoring under such conditions will be ineffectual and waste already limited network resources.

Fuzzy logic is used to implement broadcast limiting and subsequent cache management policies for entry weight assignment and entry expiration. The fuzzy system selects routes over which route discovery packets are to be rebroadcast so ultimately this system generates the paths that are to be used for data trafficking. For a static network or one consisting of slow moving nodes the fuzzy system, when cache entries expire, would tend towards discovering the same routes as those that were removed from the cache. While this in itself is a futile exercise it also restricts routing to the same subset of nodes and this can encourage localised congestion. The tuning of the fuzzy parameters for membership function crosspoints is crucial for optimising the fuzzy controller performance. Selection of unsuitable membership function ranges could cause the fuzzy broadcast limiting scheme to regard all presently existing links as being unsuitable thus leading to the discovery of no routes and effectively isolating nodes. Conversely, if the ranges are too broad then all routes may be selected making the fuzzy selection futile.

For the merging node awareness strategy, fuzzy logic is also used and again due consideration must be given to the tuning process. The objective of merging node awareness is to provide a merging network node with a functioning cache so that route discovery floods are prevented or at least reduced. This scheme is useful only if the route records that the merging node receives are paths to the nodes that it wishes to communicate with. Using this scheme in a network with a rapidly changing topology, as a consequence of high mobility, could lead to a merging node receiving stale cache
Chapter 3 Network Management Architecture

contents unless an effective cache management policy is also used to ensure cache entry reliability.

3.7 Baseline Protocol Selection for Architecture Evaluation

To evaluate the performance of the proposed overlay management architecture, the route management strategies presented were applied to DSR. This routing protocol was selected as it is in the process of being standardised by the IETF MANET working group [17], with a comprehensive protocol description detail listed in the IETF internet draft [143] which allows for accurate protocol modelling, and has been extensively evaluated in the research literature. DSR like AODV is a well-established and popular choice of routing protocol against which researchers have chosen to benchmark other protocols and strategies. These factors have influenced the decision to select DSR as the baseline protocol for the management framework evaluation. As an alternative to this AODV could easily have been selected, as it like DSR is in the process of being standardised by the IETF and again has been thoroughly evaluated in the research literature. Other routing protocols, such as LBAR and DLAR, have been shown to outperform DSR and this begs the question as to why these protocols have not be chosen as the baseline protocol for evaluation, protocols such as these are presented in the research literature without comprehensive details as to the implementation and functionality of the routing protocol. This lack of complete knowledge of the routing protocol and parameter settings will not allow for accurate modelling and can give rise to misleading simulation results.

3.8 Benchmark Testing – A Necessity

Presently, the testing and evaluation of MANET routing performance is implemented subjectively. Although, throughout the available research literature common network parameters, such as throughput, delay, and control traffic overhead, are used to evaluate network performance, the tests used are dependent on the researcher’s design and parameter selection. This is prohibitive to comparing routing strategy performance
across several proposed techniques. To realise a suite of tests that will facilitate the evaluation of different routing techniques benchmark testing is necessary. The use of benchmark testing will provide a performance basis for estimating the capabilities and limitations of MANET routing techniques and protocols. A relevant benchmark test suite must be suitable for assessing, contrasting and comparing different routing methods.

At present the IETF MANET working group [17] are involved in standardising MANET routing protocols. However, there is no such move towards developing benchmark performance evaluation tests. Ad hoc networks and in particular routing over such networks are attracting avid research attention, but due to the lack of benchmark evaluation it is difficult if not impossible to compare and contrast works as evaluation tests along with performance indices are, as previously stated, subjectively defined. The present evaluation approach is to compare proposed routing methods and algorithms against the IETF MANET standards, which comprehensively define the routing protocol functionality, as is done in this thesis. To redress this approach, benchmark testing must be developed for MANETs to push forward the development of these routing protocols. Benchmarking will make it possible to compare and contrast different approaches as published works would have to conform to a benchmark evaluation.

Introduced in [106] is a description and presentation of the actuator benchmark used in fault diagnosis studies as part of the DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) European Research Training Network. The benchmark relies on a complete set of real and stimulated scenarios for actuator fault detection and isolation. The benchmark objective is to encourage research involvement and development, knowledge dissemination and the evaluation of fault detection methods. Adopting a similar benchmark for MANET routing protocol and strategy evaluation will foster a cohesive research effort in this emerging technology. The lack of benchmark testing for protocol performance evaluation makes it impossible to gauge the level of improvement as a result of tweaking/modifying protocols or comparing protocols, the i.e. the percentage improvement in a performance metric cannot be quantified.
3.9 Conclusion

This chapter has presented the proposed dynamic route management architecture that has been developed for use with MANET routing protocols. The principles of fuzzy logic and policy based networking are used to design route management strategies that utilise multimetric channel and node-self observations. A MANET routing protocol must adapt to dynamic network conditions, to facilitate this malleability the channel and node observations are used to infer the current network state. This knowledge is used in conjunction with the network layer routing protocol to influence local routing decisions. Channel monitoring enables a network node to perceive its local neighbour activity and the quality of the links available in its one hop neighbourhood. Using self-awareness a node can deduce its suitability as an intermediate data forwarder for its local one hop set. Both fuzzy logic and policy based networking have been used to develop diverse network management techniques and a review of some of these applications has been presented with the main conclusions of these works being outlined.

Fuzzy logic has been used to develop a combined approach towards broadcast limiting and stable routing that limits redundant rebroadcasts and promotes the generation of stable routes in the route discovery process. For reactive routing protocols, this flood limiting and stable route generation algorithm is extended and used to assign weights to route table or cache entries for path differentiation that are indicative of path quality. Based on a path weighting a cache entry expiry policy has been proposed that determines expiry times using the link characteristics over which information has been received. The concept of merging node awareness has been proposed to integrate new nodes with an existing network. This scheme aims to provide current route information to merging nodes in a controlled manner, to lessen the control traffic bursts associated with merging nodes having non-existent caches. This method allows existing network nodes to determine their suitability as a next hop neighbour using fuzzy logic.

The principles of policy based networking are adopted to implement a distributed policy based management algorithm for congestion control. Network nodes monitor their congestion status and if a node is nearing a congested state or is presently congested it takes steps to alleviate this condition by notifying its immediate neighbourhood of its
Chapter 3 Network Management Architecture

congested state. The local neighbourhood responds by attempting to reroute traffic away from the congested node thus easing the node’s congestion state.

The architecture presented in this thesis is not presented as a solution that will be applicable to all network topologies, environments and operating conditions. It is a general approach to route management for MANETs that can be successfully used with a range of existing routing protocols over a range of network scenarios for enhancing network performance.

A performance evaluation of the proposed dynamic route management architecture is presented in chapter 5. This evaluation is based on the application of the route management strategies to DSR, which is a unicast topology based reactive routing protocol.
4.1 Introduction

Time or cost constraints can make it impossible to build a real system to test and measure its performance. Many real world systems are mathematically intractable and cannot be analytically assessed therefore computer simulation must be used to investigate these complex systems. Computer simulation is a suitable tool for the modelling and assessment of innovative networking concepts, prior to real-world deployment. In this thesis computer simulation is used to evaluate the performance of the proposed dynamic route management strategies for mobile ad hoc networks. Simulation allows the use of repeatable network topologies, mobility patterns and traffic flows that demonstrate the effect of network parameters [107] on system performance. Stochastic discrete event simulation is a suitable technique for modelling mobile wireless ad hoc networks as the functioning of an ad hoc network can be likened to a random process as a consequence of node mobility, traffic flows and user density. As discrete event simulation manages events in time it is used to effectively model the flow of packets in a network.

In this thesis computer simulation has been used to emulate an ad hoc network topology and this chapter describes the constituent models used to build the simulation environment. The complexity of the models used for simulation design is an important consideration. Using crude modelling will lead to crude results but large and detailed models can be difficult to implement or control and can have excessively large simulation times. To simulate real world scenarios realistic, comprehensive and accurate modelling must be used, as the significant characteristic of a real system must be captured in the model so that the simulation performance accurately portrays the actual system performance.
Chapter 4 Simulation Environment

4.2 Simulation Model Assumptions

The simulation model used for the work presented in this thesis is based on the following assumptions:

- The network consists of nodes initially positioned at points in a given topology
- The nodes move according to a particular mobility model over fixed paths in the network topology
- All nodes possess the same functionality but operate independently of each other
- Nodes access a wireless channel according to a medium access scheme
- Nodes cannot transmit and receive simultaneously
- There is no turn around time between transmit and receive modes
- For physical sensing, the channel is deemed busy if a signal above the carrier threshold is detected, with this threshold being set to 10dBs in the simulation model

4.3 Physical Layer Modelling

The simulation environment assumes a packet on the channel is detected if the SNR $\geq 10$dBs, with the noise floor assumed to $-115$dBs. This thresholding is used in conjunction with the medium access protocol to prevent nodes from simultaneously transmitting. However, when multiple signals are present on the channel a node captures the strongest signal provided the SNR threshold is satisfied and that the strongest signal is also 10dBs above the other signals present [108].

4.4 Medium Access Protocol – IEEE 802.11b DCF

IEEE 802.11 Wireless Local Area Network (WLAN) medium access control (MAC) protocol [109] is at present the dominant technology for wireless access for mobile computing devices as it provides high speed data services at a low cost. IEEE 802.11b is the most widely adopted standard and can provide up 11Mbps. In conjunction with the offered high speeds 802.11b networks operate in the 2.4GHz license free ISM
band [110]. The IEEE 802.11 specification [109] provides for three types of physical layer: direct sequence spread spectrum (DSSS), frequency hopping spread spectrum (FHSS) and Infrared (IR). In particular, the DSSS 802.11 specification was extended to 802.11b to support data rates of 5.5Mbps and 11Mbps while being backward compatible with the 802.11 DSSS specification, which supports data rates of 1Mbps and 2Mbps. The IEEE 802.11 MAC protocol tenders two modes of operation: the point coordination function (PCF) and the distributed coordination function (DCF). The PCF mode of operation provides contention free access whereas the DCF mode employs carrier sense multiple access with collision avoidance (CSMA/CA). As 802.11b DSSS based radio transmission and DCF mode is at present the most widely adopted wireless access protocol in WLANs [107,111], it was used in the simulation model environment and is discussed in the following.

4.4.1 Distributed Coordination Function

The DCF protocol uses CSMA/CA to share the medium between multiple nodes with nodes required to listen for other node transmissions, as shown in Figure 4.1. If the channel is idle then a node may transmit. However if it is perceived to be busy, nodes must wait until transmission ceases, and then instigate a random back off process. This is to preclude multiple nodes from capturing the medium directly after conclusion of the previous transmission. A prerequisite for DCF packet reception is an acknowledgement (ACK) provision. The time period between packet transmission conclusion and the commencement of an ACK frame transmission is one Short Inter Frame Space or SIFS, as per Figure 4.2. However, other packet transmissions must interpret the channel to be free for at least one DCF inter frame space (DIFS) before they are allowed to transmit.

The transmission wait time after a DIFS interval, known as the backoff period, is slotted and a node can transmit only at the beginning of each slot time. After waiting for DIFS a transmitting node enters a random backoff cycle by selecting a random integer number of slot times, from a range specified by the current contention window (CWS) size, to wait before attempting to transmit. After successfully waiting for an uninterrupted DIFS interval the backoff timer begins to decrement. If this timer counts to zero then a
node is eligible to transmit. However, if the channel is captured by another node before the backoff timer has expired the backoff process is suspended, with the current backoff time being recorded and will be used in succeeding transmission attempts until the backoff timer reaches 0. Selection of the random backoff interval is done using the CWS value. The backoff period is drawn as a pseudorandom integer from a uniform distribution over the interval \([0, \text{CWS}]\), where \(\text{CW}\) is an integer within the range of values of the physical layer characteristics \(\text{CWS}_{\text{min}}\) and \(\text{CWS}_{\text{max}}\). \(\text{CWS}\) increases exponentially when a packet transmission fails until it reaches \(\text{CWS}_{\text{max}}\) where it remains and is reset after a successful packet transmission or the packet retry transmission limit has been reached.

![DCF Access Contention](image)

**Figure 4.1** DCF Access Contention

The backoff period attempts to prevent packet collisions by using physical carrier sensing on the assumption that all nodes are within communications range of each other.
Chapter 4 Simulation Environment

However, this is not a valid premise as shown in Figure 4.3. Node A is within transmission range of Node C but out of range with respect to node B therefore it would not ‘hear’ transmissions from B to C by merely sensing the channel. Consequently the probability of packet collisions is greatly increased. This is referred to as the hidden terminal problem [112]. The solution to this problem is virtual carrier sensing. Virtual carrier sensing allows a node to reserve the channel for a precise interval of time through the use of request-to-send (RTS) and clear-to-send (CTS) frames. For the scenario shown in Figure 4.3 node B transmits an RTS frame to node C. As node A is out of range this RTS is not detected. A part of the RTS frame header is a duration field that states the period of time for which the channel is to be reserved for an ensuing packet transmission. Nodes overhearing this RTS frame update their network allocation vector (NAV) with this medium reservation information. When node C receives the RTS frame it responds by sending a CTS frame, as shown in Figure 4.3. The CTS frame also contains transmission duration information. As node A is within range of node C it updates its NAV with the CTS duration information and will not attempt to transmit. This RTS/CTS action prevents possible collisions between hidden terminals.

![Figure 4.3 Hidden Terminal Problem](image)

When a frame arrives at the MAC layer from an upper layer it is encapsulated in a MAC protocol data unit (MPDU) and is sent to the physical layer where the physical layer convergence protocol (PLCP) header and preamble are attached, as per the frame format shown in Figure 4.4 [109]. The PLCP is transmitted at a rate of 1Mbps and takes 192µs to transmit whereas the MPDU is transmitted at a rate equal to or above the PLCP
transmission rate (2, 5.5, 11Mbps). Table 4.1 shows the DSSS physical layer characteristic values [109] that have been used in the simulation model.

<table>
<thead>
<tr>
<th>MPDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCP Preamble</td>
</tr>
<tr>
<td>PLCP Header</td>
</tr>
<tr>
<td>MAC Header</td>
</tr>
<tr>
<td>Frame Payload</td>
</tr>
<tr>
<td>FCS</td>
</tr>
<tr>
<td>Bytes:</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>≤2304</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

**Figure 4.4** PLCP Protocol Data Unit Frame Format

**Table 4-1 DSSS Physical Layer Characteristics**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot Time</td>
<td>20μs</td>
</tr>
<tr>
<td>SIFS</td>
<td>10μs</td>
</tr>
<tr>
<td>DIFS</td>
<td>50μs</td>
</tr>
<tr>
<td>Contention Window Min, Max</td>
<td>31, 1023</td>
</tr>
<tr>
<td>PLCP Preamble</td>
<td>144μs</td>
</tr>
<tr>
<td>PLCP Header</td>
<td>48μs</td>
</tr>
</tbody>
</table>

**4.5 Packet Reception**

When a node receives a packet, it needs to determine whether or not it can decode the packet correctly. The probability of a successful transmission [113] is calculated using the received SNR as follows:

Assuming a physical layer rate of 11Mbps and a frame length of $L$ bytes, then the probability of a successful transmission is given by:

$$P_{\text{success}}(L) = (1 - P_{e_{-d}}(L)) \cdot (1 - P_{e_{-a}})$$

where $P_{e_{-d}}(L)$ and $P_{e_{-a}}$ are the probability of error for a frame of size $L$ bytes and an ACK frame, respectively. As an ACK frame is transmitted at a rate equal to or less than
Chapter 4 Simulation Environment

the data frame rate with a relatively small size of 14 bytes [109] when compared to the
data frame size its associated error probability is much lower than the data frame, thus,
allowing 4.1 to be approximated to:

\[ P_{\text{success}}(L) = (1 - P_{e\cdot d}(L)) \]  

4.2

The error probability associated with an \( L \) byte data frame is:

\[ P_{e\cdot d}(L) = 1 - \left(1 - P_e(24)\right) \cdot \left(1 - P_e(28 + L)\right) \]  

4.3

where \( P_e(24) \) is the probability of error associated with the PLCP preamble and header
transmission at a rate of 1Mbps and \( P_e(28+L) \) is the MPDU error probability, as per
Figure 4.4. Now, \( P_e(L) \) can be written in terms of the bit error rate (BER) \( P_b \) as:

\[ P_e(L) = 1 - \left(1 - P_b\right)^{2L} \]  

4.4

with the BER \( P_b \) being estimated using the received SNR and the empirical curves shown
in Figure 4.5 [114]. In the simulation environment when a node receives a packet it
calculates the probability of success using equation 4.2 and draws a uniform random real
number, \( R_x \) from the interval \([0,1]\), if \( R_x < P_{\text{success}} \) then the node assumes that the packet
can be decoded and processes it otherwise the packet is dropped.

Figure 4.5 IEEE 802.11b BER vs. SNR
Chapter 4 Simulation Environment

4.6 Radio Propagation

The radio channel between transmitting and receiving nodes presents a hostile environment for data transmission and so must be considered in the computer simulation model environments. To determine the signal power loss between transmitter and receiver the attenuation due to distance, shadowing and obstacle interference must be considered [115]. The radio propagation model used in this simulation environment considers path loss due to distance, shadowing and obstacle interference.

4.6.1 Path Loss

The path loss determines how much the mean received power decreases with distance. The attenuation due to distance is based on free space loss and is calculated using the Friis transmission formula [115, 116]:

\[
\frac{P_r}{P_t} = G_t G_r \left( \frac{\lambda}{4\pi r} \right)^2
\]

where \( P_t, P_r \) are the node transmit and receive powers, \( G_t \) and \( G_r \) are the gains of the transmitting and receiving node antennas, \( r \) is the distance between the nodes and \( \lambda \) is the wavelength. For the sake of simplicity the antenna gains, \( G_t \) and \( G_r \) are assumed to be unity. The simulation topologies used, as discussed in section 4.9, consist of buildings so obstacle interference is seen as the intersection of a communications path between transmitter and receiver nodes with these buildings. For each wall that intersects this path a loss of 15dBs is considered [115].

4.6.2 Shadowing

Shadowing or slow fading is modelled by a log-normal distribution with zero mean and standard deviation \( \sigma \), with \( \sigma \) being set to 10dBs for an outdoor environment [115, 116]. The probability density function is:
Chapter 4 Simulation Environment

\[ P(L_s) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{L_s^2}{2\sigma^2}\right), \] \(L_s\) is the loss due to shadowing in dBs

As shadowing is a spatially correlated process [117, 118] there must be some correspondence between the shadow values at nearby topological positions, which must be considered in the simulation environment. Presented in [117] is a detailed analysis of a method for correlating a shadow sequence and this method was adopted here. The model filters an uncorrelated shadow sequence through a filter with an appropriate impulse response that leads to an output sequence with the required correlation. The filter impulse response necessary for the generation of the required correlation is derived in [117] and is given as:

\[ h[n] = \sqrt{1 - e^{-2\alpha}} e^{-\alpha n}, n \geq 0, \alpha = \ln 2 \]

This filter exhibits an infinite impulse response, as per Figure 4.6 [117], but truncating the filter response at \(n = 6\) produces a good approximation. This filter was implemented in the shadow model used in the simulation environment to correlate a sequence of uncorrelated log normally distributed samples with a mean of zero and a standard deviation of \(\sigma\). The simulation environment network nodes draw an initial shadow value at simulation start-up.

![Correlating Filter Impulse Response](image-url)
Chapter 4 Simulation Environment

4.6.3 Signal to Noise Ratio

The received signal strength $P_{rx}$, in dBm, is calculated as:

$$P_{rx} = P_t - \text{PathLoss}$$  \hspace{1cm} 4.8

where $P_t$ is the transmit power set at 10dBm and the PathLoss is given by:

$$\text{PathLoss} = LD + LS + LO$$  \hspace{1cm} 4.9

where $LD$ is attenuation due to distance, $LS$ is the shadow value and $LO$ is the obstacle interference, all units are in dBs.

The SNR in dBs is calculated as:

$$SNR = (P_{rx} - 30) - \text{NOISE}$$  \hspace{1cm} 4.10

where NOISE is the is the noise power [115] given by:

$$\text{NOISE} = kTB$$  \hspace{1cm} 4.11

where $k$ is Boltzmann’s constant, $T$ is the absolute temperature in Kelvin and $B$ is bandwidth. For the simulation model presented NOISE is set to –115dBs.

If,

$$SNR \geq SNR \_\text{Threshold}$$  \hspace{1cm} 4.12

then a node assumes it can ‘hear’ a transmission ($SNR \_\text{THRESHOLD} = 10$) and using 4.2 a node determines whether or not it can successfully decode a packet.

4.7 Traffic Flows

Simulation traffic flows are modelled as continuous bit rate (CBR) traffic with a fixed packet size of 512 bytes. Approximately 30% of the total number of network nodes act as CBR sources with exponentially distributed activity intervals with an average duration of 5 seconds. The packet generation rate per second is varied so as to alter the network load conditions. The proposed network management strategies are concerned with routing at the network layer and thus are not interested in the applications generating data traffic but rather the successful routing of those packets. Consequently, data traffic assumes UDP transport control. As this is best effort no transmission retries are attempted at the transport layer.
4.8 Battery Model

A linear battery model [119] is assumed so the battery is regarded as a linear storage of current with the maximum battery capacity being realisable. The remaining battery capacity, $RC$, after a transmit/receive operation of duration time $t_d$ is given by:

$$RC = C - \int_{t=t_0}^{t=t_0+t_d} I(t)dt$$

where, $C$ is the previous capacity, $I(t)$ is the instantaneous current consumed by the transmit/receive operations at time $t$. It is assumed that the current is constant for the duration $t_d$. This simplifies 4.13 to:

$$RC = C - I \cdot t_d$$

Simulation nodes are assigned an initial battery capacity up to a maximum battery capacity and are depleted according to equation 4.14 and the simulation parameters shown in Table 4.2.

<table>
<thead>
<tr>
<th>Table 4-2 Linear Model Battery Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Capacity</td>
</tr>
<tr>
<td>Transmit Operation</td>
</tr>
<tr>
<td>Receive Operation</td>
</tr>
<tr>
<td>Supply Voltage</td>
</tr>
</tbody>
</table>

4.9 Simulation Test Scenarios

Four environment models were simulated, as per Figures 4.7 - 4.10, with respective areas of 50m x 20m, 200, 500 and 1000 square metres. These areas consist of buildings, represented by green rectangles, positioned over the available area. The white areas constitute open outdoor unobstructed spaces. To generate realistic mobility patterns obstacles must be included in the environment so that voronoi polygons can be constructed and mobility patterns can be derived. The environment models presented were designed with a view to generating realistic node movement and incorporating obstacle inference in the radio propagation model. The topology area and obstacle density
Chapter 4 Simulation Environment

used in the environments were selected to test the proposed multimetric performance over varying simulation environments.

Figure 4.7 50m x 20m Topology

Figure 4.8 200m x 200m Topology

Figure 4.9 500m x 500m Topology
Figure 4.10 1000m x 1000m Topology

4.10 Mobility Modelling

A mobility model must endeavour to imitate realistic movement patterns [120]. When nodes have been initially dispersed over an area the mobility model directs the movement of nodes in the network. The performance of the routing protocol is strongly correlated with the accuracy of the mobility model. Simulation results extracted using an unrealistic mobility model may give rise to a false impression of the routing protocol performance. Network simulation involving movement patterns are implemented using either mobility traces or synthetic models. Mobility traces are used to capture real world movement whereas synthetic models strive to realistically imitate node movement without the application of traces. Traces can be used to model movement over existing
networks whereas synthetic models must be used to derive mobility patterns for new network environments, such as ad hoc networks. A synthetic mobility model must generate node movements that are akin to realistic movements. Nodes must move with varying velocities, direction and pause for arbitrary time periods before continuing. Currently, there exists several synthetic mobility models for ad hoc networks such the Random Walk, Random Waypoint, Random Direction, Boundless Simulation Area, Gauss-Markov, Probabilistic Random Walk and the City Section mobility models [121].

Typically the Random Waypoint mobility model is used for mobility modelling for MANET routing protocol evaluations and represents movement of a node within a bounded area, $A$. Initially nodes are uniformly positioned over $A$. Each node then selects a destination from a uniform distribution over $A$ and the node moves along a straight line from source to destination with a constant velocity drawn from a uniform distribution independent of the node’s position. When a node reaches its destination it pauses for a certain time period and the process repeats. However, this model and others like it, such as the Random Walk and Random Direction models, are memoryless models and do not produce realistic mobility [120, 121, 122]. With the Random Walk and Waypoint models, nodes use straight-line movement between arbitrarily selected destination points, which does not represent true-life movement. The Random Direction model produces unrealistic trace patterns as it encourages mobile nodes to move towards the simulation area boundary and pause there before moving towards another edge.

The Boundless Simulation Area mobility model [121] maintains a relationship between a node’s previous velocity and direction and its current velocity and direction. Nodes using the previous mobility models are forced to stop or bounce off a simulation boundary edge whereas nodes moving according to the Boundless Simulation Area model simply reappear on the opposite side of the simulation area. This results in nodes moving over a torus-like topology. This model provides unobstructed mobility however, if there exists stationary or relatively slow moving nodes then fast moving nodes travelling repeatedly in a similar direction can constantly become neighbours with these nodes, which is unrealistic and it can lead to a static topology.

The Gauss-Markov model [121] updates a node’s speed and direction at fixed time intervals. The new speed and direction values are determined using the previous
Chapter 4 Simulation Environment

speed, direction values and a tuning parameter that is used to vary the randomness. This model can produce real world mobility patterns but this depends on the selection of an appropriate model tuning parameter. The Probabilistic Random Walk model uses a probability matrix to determine the position of a node where each matrix entry $P(a,b)$ represents the probability that a node will go from state $a$ to $b$. This model produces probabilistic movement rather than random movement, which may produce more realistic movement patterns, but this model, like the Gauss-Markov model, requires the selection of an appropriate transition probability matrix.

The City Section mobility model [121] uses a street network to represent an actual city topography where an ad hoc network exists or is to be deployed. Nodes move between streets along fixed pathways. This model does produce realistic mobility patterns as it constrains nodes to moving along specified pathways. However, in order to use this model a real life topography is desirable and nodes should be allowed to move through buildings when moving between streets as people typically take the shortest most direct path between destinations.

The Obstacle Mobility Model presented in [122, 123] incorporates movement along restricted fixed pathways and around and via obstacles thus allowing the modelling of movement over realistic topographies. This model provides a mechanism of constructing pathways around and through obstacles using Voronoi polygons [122, 123, 124] thus allowing a more realistic mobility pattern to be realized than those models previously discussed, which simply considered movement over a flat unobstructed topological area. The obstacles placed in the bounded $x,y$ area can represent buildings or other structures that restrict node movement to fixed pathways and act as a possible barrier to node communications.

The pathway movement graph is created using a Voronoi diagram. Mathematically a Voronoi diagram is described as the partitioning of a plane with $n$ points into $n$ convex polygons, known as voronoi polygons, such that each Voronoi polygon contains exactly one generating point and every other point in a given polygon is closer to its central point than to any other [122, 123]. The set of $n$ points are known as location points. For the Voronoi based obstacle mobility model the $x,y$ coordinates of the corners of the obstacles (i.e. buildings) are used as location points around which the
Voronoi polygons are constructed. The vertices of the Voronoi polygons and bounded simulation area provide the pathways with the intersection of a Voronoi vertex and an obstacle edge being used as a means of traversing a building i.e. doorways. The set of available node positions or sites are derived from the points of intersection of the Voronoi vertices with the simulation area boundary and obstacles. Initially nodes are uniformly scattered over these sites and move via these sites by uniformly selecting a destination site and determining the shortest path between the source and destination site. The velocity and pause times at destinations sites are, like the random waypoint model, selected using uniform distributions.

As people typically move along the shortest path between locations, nodes using the Obstacle mobility model construct the shortest path between sites by considering the Voronoi diagram as a graph of undirected edges [125] with the weight of each edge being the Euclidean distance between the edge vertices. When a node selects a destination site it runs a shortest path finding algorithm, such as Dijkstra [125], to determine the shortest path between it and the destination using the Voronoi site positions to create an adjacency matrix from which the shortest path can be generated from[126]. The topologies as per Figures 4.7 – 4.10 are used to create mobility pathways for the simulation environment. The corresponding Voronoi diagrams are shown in Figures 4.11 – 4.14 with the building corner points being plotted, which are labelled with numbers, and used as the set of n location points, as per Table 4.3. These figures show the Voronoi polygons, constructed about each numbered location point, that generate the movement graph for mobility simulation. The number of Voronoi site positions generated for each topology is shown in Table 4.4.

<table>
<thead>
<tr>
<th>Topology</th>
<th>Set Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>50m x 20m</td>
<td>60</td>
</tr>
<tr>
<td>200m x 200m</td>
<td>40</td>
</tr>
<tr>
<td>500m x 500m</td>
<td>40</td>
</tr>
<tr>
<td>1000m x 1000m</td>
<td>80</td>
</tr>
</tbody>
</table>
### Chapter 4 Simulation Environment

#### Table 4-4 Voronoi Diagram Site Set Size

<table>
<thead>
<tr>
<th>Voronoi Diagram</th>
<th>Set size</th>
</tr>
</thead>
<tbody>
<tr>
<td>50m x 20m</td>
<td>121</td>
</tr>
<tr>
<td>200m x 200m</td>
<td>70</td>
</tr>
<tr>
<td>500m x 500m</td>
<td>74</td>
</tr>
<tr>
<td>1000m x 1000m</td>
<td>133</td>
</tr>
</tbody>
</table>

#### Figure 4.11 50m x 20m Topology Voronoi Diagram

#### Figure 4.12 200m x 200m Topology Voronoi Diagram
Chapter 4 Simulation Environment

Figure 4.13 500m x 500m Topology Voronoi Diagram

Figure 4.14 1000m x 1000m Topology Voronoi Diagram
Chapter 4 Simulation Environment

4.10.1 Mobility Model Effect on Simulation Performance

As stated previously the choice of mobility can have a significant effect when evaluating protocol performance [121, 122]. Consequently, an appropriate mobility model should be selected for performance appraisal [122, 123]. To demonstrate this effect, test simulations were executed using the random waypoint model, with boundary wraparound to remove edge effects, and the Voronoi Polygon Obstacle model over a 200m x 200m topology. The Random Waypoint considers the topology as a flat area, disregarding obstacles, and simply moves in a straight line between random source and destination points. The Obstacle model however, constructs pathways around and via these structures, as shown in Figure 4.12, so nodes are restricted to movement over these fixed paths between specific site locations. Shown in Figure 4.15 is a sample of the mobility pattern for an individual node for the Random Waypoint mobility model run over a 200m x 200m topology.

![Random Waypoint Mobility Pattern For An Individual Node](image-url)

**Figure 4.15** Random Waypoint Mobility Pattern For An Individual Node
Node’s using the Obstacle mobility model for the 200m x 200m topology shown in Figure 4.8 move over the area using the Voronoi polygon based movement graph derived in Figure 4.12. These node movement patterns are very different with the Obstacle mobility pattern being more reasonable as it can be applied to realistic topographies as it considers movement around and via structures whereas the random waypoint considers only straight line between source and destination points. The effect of mobility model selection on the performance of the DSR routing protocol performance with 10 nodes is presented in Figure 4.16, with each of the metrics shown being normalised using the largest value for each category as the normalising value. From Figure 4.16 it is evident that the selection of the Obstacle mobility model causes a poorer protocol performance as the packet delivery ratio is reduced, more control traffic is generated and a larger number of attempted route requests are aborted. However, this mobility model gives more realistic results and facilitates a clearer evaluation of the routing protocol performance as movement is restricted and structure interference is considered in both the mobility model and the propagation model.

![Figure 4.16 The Effect Of Mobility Model Selection On Protocol Performance](image)
Chapter 4 Simulation Environment

4.11 Simulation Model

A simulation tool has been designed to provide a modular application environment where specific, user defined scenarios can accurately be modelled for the evaluation of the proposed route management strategies. This simulator models an IEEE 802.11b WLAN in ad hoc mode and the simulation model structure is shown in Figure 4.17. The simulation model was developed by the author using the models described previously and is implemented in C++ using the stochastic discrete event simulator Communication Network Class Library (CNCL) [127]. The choice to develop this simulator and not rely on commercially available tools such as OPNET, or freely available simulators like NS2, was taken so that complete control could be exercised over the model development allowing for comprehensive modelling. By developing this simulator a complete knowledge of the system is gained allowing for control over and a better understanding of the system parameters and interaction between the system layers, such in-depth knowledge is not achievable with tools that have been procured. For network simulation the system is modelled so that the performance of the system can be expressed in terms of generating and transmitting events, with events being likened to packets. Network node movement patterns are derived using the Voronoi Polygon Obstacle mobility model for the topology scenarios shown in Figures 4.11 - 4.14.

4.12 Random Number Generation & Statistical Analysis

The modelling of random behaviour for simulation purposes is implemented via stochastic processes. CNCL incorporates pseudorandom number generators [116, 128] for stochastic process simulation. For the simulation environments developed in this study a Lagged Fibonacci random number generator, available with CNCL, has been used as the base generator for all simulations. Such a pseudorandom number generator has been shown to exhibit good statistical properties with regard to correlation of generated random number sequences and has a large period of $(2^{32}-1)*2^{96}$ before the sequence repeats [116].
The statistical evaluation of simulation results is necessary in establishing how well the simulated model captures the true behaviour of the system. To provide a good estimate of the observed system characteristics, confidence intervals for the estimated expected values of the output processes are evaluated. The Batch Means method [128] has been used in conjunction with the simulation scenarios presented in chapter 5 in determining confidence intervals for the sampled output random values. The batch means method samples values from a single simulation run of length $m$ and divides the resulting
observations into \( n \) batches of length \( k \), with \( m = nk \), and statistical evaluation is independently performed on each batch. The batch mean parameters \( n \) and \( k \) are appropriately chosen so as to achieve a good estimate of system characteristics, with \( k \) typically being: \( 10 \leq k \leq 30 \) [129]. The mean relative error of the observed values was less than 0.05 for all cases.

4.13 Conclusion

Computer simulation is necessary in analysing mathematically intractable systems and is used to investigate system performance prior to real-world deployment. This chapter has discussed computer simulation modelling and described the constituent models used to implement the stochastic discrete event ad hoc simulator developed as part of this study. When designing a computer simulation environment accurate modelling must be used, as crude system modelling will not capture the significant characteristics of the real system and will generate misleading performance evaluations.

Realistic mobility modelling is critical for MANET routing protocol evaluation as the performance of a routing protocol is strongly correlated with mobility. The random waypoint mobility model is normally the preferred mobility model when evaluating MANET routing protocol performance, however, the mobility patterns generated are straight-line movements between random source and destination points and are not representative of realistic movement. To generate mobility patterns that are reflective of realistic movement the mobility model implemented as part of the developed simulator is the Voronoi Polygon Obstacle. This model considers movement around and via obstacles. The simulation scenarios used to evaluate the proposed route management strategies consist of buildings positioned in an \( x, y \) area and the Voronoi polygon mobility model generates pathways around and via these buildings. Such a mobility model restricts nodes to movement over specific paths. However, the mobility patterns generated are more realistic than those produced by models like the random waypoint mobility model. Consequently, the system performance achieved with such a model is more realistic than a performance based on random waypoint mobility.
Chapter 4 Simulation Environment

The following chapter presents simulation evaluations of the proposed dynamic route management strategies for mobile ad hoc networks with the simulation model presented in this chapter being used to extract the performance results presented in chapter 5.
Chapter 5 Management Architecture Evaluation

5.1 Introduction

This chapter presents a computer based simulation evaluation of the proposed route management strategies discussed in chapter 3. For the purposes of demonstrating the effectiveness of the developed dynamic route management strategies the corresponding algorithms are incorporated with the reactive unicast topology based protocol DSR to exhibit how network performance is enhanced as a consequence of these methods. The proposed management framework could however be also applied to other on demand routing protocols, such as AODV, and equally to other classes of routing protocols where it can be used as an extension to existing routing policies, as was discussed in chapter 3. Due to computationally extensive and excessive simulation runtimes the number of network nodes used in scenarios was kept deliberately low.

5.2 Congestion State Monitoring using Distributed Policy

The effectiveness of the distributed policy congestion monitoring approach is investigated using average end-to-end data traffic delay measurements in conjunction with queue performance and channel retransmissions. The congestion level of a node varies over simulation time, as shown Figure 5.1, with levels varying from 1 (not congested) to 5 (high level of congestion). As expected the number of congestion warnings and notifications sent rises as both network density and traffic load increase for a given topology, with an example of the total number of \( C_W \)s and \( C_N \)s generated for all nodes being shown in Figure 5.2. If a node determines itself to be in a congested state it sends either a \( C_W \) or \( C_N \). The purpose of these broadcasts is to notify the local neighbourhood of the nodes current level of distress and to direct traffic away from this node, if possible. Consequently, the congested node should find the channel around it...
becomes less busy, with the node’s queue and transmission performance improving after either a $CW$ or $CN$ has been broadcast and the overall network end-to-end data traffic delays should be improved.

![Figure 5.1 Sample Individual Node Congestion Level during a simulation run](image)

![Figure 5.2 Congestion Event Occurrence for varying node density](image)

### 5.2.1 Scenario Simulation Results

The following sections display simulation results for the congestion monitoring scheme as applied to network topologies of 200, 500, 1000 meters squared respectively and a topology with an area of 50 by 20 meters, as shown in Figures 4.7 – 4.10. The
Chapter 5 Management Architecture Evaluation

Simulation experiments were implemented for constant mobility and for mobility with random pause times drawn from a uniform distribution with a range of 10 – 180 seconds using the Voronoi Obstacle mobility model discussed in chapter 4. The simulation analysis presented in section 5.2.1.1 corresponding to the 200 meters squared topology shows results for constant mobility and mobility with pause times, however as regards simulation analysis for the 50m x 20m, 500 and 1000 meters squared topologies Table 5.5 shows the statistical performance values and percentage reduction achieved with the congestion monitoring approach for scenarios that have constant mobility as the simulation results for mobility with pause times are comparable.

5.2.1.1 Topology – 200 Meters Squared

Setup: Number of Nodes: 10, with constant mobility

Presented in Figure 5.3 are plots that show the mean queue state values, as per equation 3.2 (denoted as Queue Ratio in the Ratio Metric Statistics Tables) and transmission state values, as per equation 3.3 (denoted as Transmission Ratio in the Ratio Metric Statistics Tables) averaged over the complete simulation time for each node in the network and are indicative of the network interface queue performance and packet retransmission ratio. The purpose of the congestion monitoring approach is to improve node queue performance and lessen retransmissions using reactive congestion control to distribute traffic away from congested nodes. As stated in chapter 3, a node measures its queue state as the ratio of the number of packets that were presented at the network interface queue but were dropped, as the queue was full, to the number of packets that were actually placed in the queue, in a 1-second interval. The transmission state is the ratio of the number of retransmissions to successful transmissions (not including broadcast packets), in a 1-second interval. For these metrics the lower the ratio value the better the performance achieved. Table 5.1 compares the overall system mean and standard deviation values for the average queue state and transmission state ratios shown in Figure 5.3. With congestion monitoring the mean and standard deviation values for the overall system queue and transmission ratios are lower than the non-monitored method.
Chapter 5 Management Architecture Evaluation

These lower mean values indicate that the overall average queue and transmission performance is better than the non-monitored system. The standard deviation values associated with the monitored system are smaller than the non-monitored system. This obviously indicates that the mean queue and transmission ratio values for individual nodes are closer to the overall system mean values, which indicates that the network load is distributed more fairly as overall a better queue and transmission performance is achieved.

![Graph showing average ratio metrics for individual network nodes - 200 meters squared & constant mobility]

**Figure 5.3** Average Ratio metrics for individual network nodes – 200 meters squared & constant mobility

<table>
<thead>
<tr>
<th>Table 5-1 Ratio Metric Statistics - 200 meters squared &amp; constant mobility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Queue Ratio</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>With Monitoring</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
<tr>
<td><strong>Transmission Ratio</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>With Monitoring</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
</tbody>
</table>
Chapter 5 Management Architecture Evaluation

If a node determines itself to be in a congested state it sends a congestion warning, $CW$ or a congestion notification, $CN$. The aim of these congestion broadcasts is to notify the local neighbourhood of the nodes current level of distress and to direct traffic away from this node, if possible. Consequently, the congested node should find the channel around it becomes less busy, with the queue state and transmission state metrics lowering in value after either a $CW$ or $CN$ has been broadcast. This effect is demonstrated in Figures 5.4 and 5.5. Shown in Figures 5.4 is the congestion monitoring effect on the queue state metric and on the transmission state metric for an individual node, with a sample of 50 values being presented. The congestion level of a node is signified in the range $0 - 5$ with a node broadcasting a $CW$ if the congestion level $= 4$ (MED-HIGH) or a $CN$ if the congestion level $= 5$ (HIGH). If the congestion level is set to a value less than 4 no action is taken. When a node is in a distressed state (congestion level $= 4$ or 5) the appropriate congestion packet is broadcast. The aim of this is to recover the node to a non-congested state. From Figure 5.4, it is evident that when the congestion level falls from level 4 or 5 to level 3 or below the queue state metric also falls, which is indicative of improved queue performance.

![Figure 5.4 Congestion Monitoring Effect on Queue State Metric with constant mobility](image)

Likewise for the transmission state metric as the node congestion level falls below level 4 or 5 the transmission ratio improves, as the node is no longer congested.
Setup: Number of Nodes 10, mobility with pause times

Similar results to those previously discussed were obtained for this scenario, which employs the Voronoi obstacle mobility model with random node pause times. The consequent simulation results are shown in Figures 5.6 – 5.8 and Table 5.2. From Figure 5.6, the node with an ID of 6 has a poor average queue performance before the application of congestion monitoring, this could be attributed to the node being used by many of its neighbours as an intermediate node for data routing or that the node’s local channel is excessively busy, which prevents this node from transmitting.
Table 5.2 compares the overall system mean and standard deviation values for the average queue state and transmission state ratios shown in Figure 5.6. Similar results are obtained for this scenario, which incorporates random node pause times in the mobility model, as were concluded for the previous scenario. Again the congestion monitoring approach improves the mean system queue and transmission ratio performances as shown in Table 5.2. The effects of the $CW$ and $CN$ broadcasts are demonstrated in Figures 5.7 and 5.8.
Data Throughput with congestion monitoring is increased in the region of 3% - 8% when compared against non-congestion monitored routing. Conversely, there is a slight increase in route discovery overhead being incurred as an effect of congestion monitoring (typically in the region of 1% - 3%). However, the network average end-to-end data traffic delay for successfully delivered data traffic, facilitated by congestion monitored routing, is greatly improved when compared against non monitored routing as node queue performance and transmission performance are enhanced thus reducing network delay, as shown in Table 5.3.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>% Reduction in Average End-To-End Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant Mobility</td>
<td>20%</td>
</tr>
<tr>
<td>Mobility with Random Pausing</td>
<td>15%</td>
</tr>
</tbody>
</table>

As is evident from Table 5.3 a higher percentage reduction in average end-to-end delay is achieved when the congestion monitor approach is applied to a scenario with constant mobility. With constant mobility network nodes are aware of more routes as new links are formed at a greater rate than would be created when considering mobility with
pause times as nodes remain stationary for finite periods of time before moving towards the next destination. The congestion monitoring approach involves nodes redistributing traffic over non-congested nodes so the more routes available to a node the more likely it is that the node will be aware of alternate routes not containing currently congested nodes. This allows the diversion of traffic away from congested network zones and results in lower end-to-end delay measurements for the constant mobility case.

**Setup:** *Number of Nodes: 5, constant mobility*

The proposed Congestion monitoring scheme is at present not as effective when applied to sparsely populated networks, where the network area is large relative to the number of nodes, as demonstrated in Figures. 5.9 – 5.11 and Table 5.4, as nodes have a limited number of neighbours due to the spread of the nodes over the available network area, consequently a small number of paths are available to those neighbours. Similarly, the difference between the data throughput and control traffic overhead measurements for congestion monitored routing and non-congestion monitored routing is negligible while the average end-to-end data traffic delay for congestion monitored routing is reduced by 4%.

![Figure 5.9](image)

**Figure 5.9** Average Ratio metrics for individual network nodes – 200 meters squared & constant mobility
Table 5-4 Ratio Metric Statistics - 200 meters squared & constant mobility

<table>
<thead>
<tr>
<th>Queue Ratio</th>
<th>With Monitoring</th>
<th>Without Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>0.4831</td>
<td>0.4948</td>
</tr>
<tr>
<td><strong>Standard Deviation</strong></td>
<td>0.1198</td>
<td>0.1117</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transmission Ratio</th>
<th>With Monitoring</th>
<th>Without Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>0.0366</td>
<td>0.0347</td>
</tr>
<tr>
<td><strong>Standard Deviation</strong></td>
<td>0.0138</td>
<td>0.0149</td>
</tr>
</tbody>
</table>

As is shown in Figure 5.9, the performance of the congestion monitoring approach is not as effective for sparsely populated networks, as the mean queue and transmission ratio performance are very similar with and without the application of congestion monitoring. Likewise, in Table 5.4 this is also evident as the system mean and standard deviation values for the queue and transmission ratio metrics are comparable which indicates the ineffectiveness of the congestion monitoring approach for sparse networks. The congestion warning scheme involves nodes rerouting packets if an alternate route is available that does not contain the congested node otherwise the packets are dropped. For the congestion notification scheme packet rerouting is also attempted but unlike the warning method routes containing congested nodes are forcibly expired from the cache. As few routes are available, as a result of the low-density network, the expired routes can be rediscovered thus leading to continued poor queue and transmission performance even after a congestion warning or notification has been broadcast. With a denser network alternate routes are normally available, as route discovery packets that are sent over non-congested network zones would be returned quickly allowing the discovery of other routes while route discovery packets that are sent toward congested zones are dropped or heavily delayed. To circumvent this problem and make the congestion monitoring scheme suitable for sparse networks a congested node should not be included in subsequent route rediscoveries for a specific timeout interval therefore forcing the generation of an alternate route if it exists. Such an approach would also be beneficial in
improving network throughput, as this process would encourage the generation of better quality paths through lightly loaded network nodes that have access to the channel.

5.2.1.2 Multiple Scenarios with Constant Mobility

As the simulation plots achieved with the 500, 1000 meters squared and 50m x 20m topologies are comparable to those presented in the previous section, Table 5.5 is used to demonstrate the effectiveness of the congestion monitoring approach. Presented are the statistical performance measurements and the percentage reduction achieved in the average network end-to-end delay as a consequence of congestion monitoring.

<table>
<thead>
<tr>
<th>Area</th>
<th>NN</th>
<th>S</th>
<th>QR</th>
<th>TR</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 x 500</td>
<td>20</td>
<td>MT M</td>
<td>0.3699</td>
<td>0.2275</td>
<td>31%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SD</td>
<td>0.0621</td>
<td>0.0715</td>
<td></td>
</tr>
<tr>
<td>1000 x 1000</td>
<td>40</td>
<td>MT M</td>
<td>0.3011</td>
<td>0.3124</td>
<td>51%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SD</td>
<td>0.1498</td>
<td>0.1761</td>
<td></td>
</tr>
<tr>
<td>50 x 20</td>
<td>5</td>
<td>MT M</td>
<td>0.3833</td>
<td>0.1236</td>
<td>19%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SD</td>
<td>0.0675</td>
<td>0.0165</td>
<td></td>
</tr>
</tbody>
</table>

Table Key
Area: Topology size in meters, with constant mobility
NN: Number of nodes
S: Standard deviation and mean statistical values
QR: Queue ratio performance
TR: Transmission ratio performance
PR: Percentage reduction in average network end-to-end delay as a consequence of congestion monitoring
WM: With congestion monitoring
WOM: Without congestion monitoring
M: Mean
SD: Standard deviation
From Table 5.5 it is evident that the congestion monitoring has greatly improved the overall system performance for the scenarios tested. The overall mean for the queue and transmission ratio metrics are lower as a consequence of congestion monitoring and thus is indicative of a better average performance. Likewise, the standard deviation values associated with these metrics is smaller. Again, this is indicative of network nodes having performed comparably and so no individual node or subset of nodes is congested for sustained period of time during the simulation course.

The proposed distributed policy based reactive congestion monitoring scheme has demonstrated its effectiveness by in particular, improving the average end-to-end network delays with the percentage reduction in this metric being in the region of 15% - 51% for the test scenarios. This corresponds to a decrease in network interface queue wait times and a lessening in the need for retransmissions. The congestion monitoring scheme is ideally suited for large dense networks as is evident from the simulation results presented, as the larger more dense topologies exhibit the greatest improvement. The rationale for this can be attributed to the fact that in such scenarios more nodes are available over a larger area consequently additional routes are available for load redistribution making it easier to avoid congested network zones. This inclusion of a congested node timeout interval in the subsequent route discover process after a congestion notification broadcast will prevent congested nodes from attempting to participate in this process, allowing sufficient time for the local neighbour congestion to dissipate before traffic is again directed towards this network zone and will increase network throughput.
5.3 Fuzzy Logic Based Routing

Fuzzy logic has been used in this study in the route discovery process to promote broadcast limiting and the generation of stable routes. These routes based on their associated fuzzy metrics are assigned variable paths and dynamic route cache expiry times. With fuzzy logic assisted routing it is expected that an increased number of route requests arrive at the necessary destination as these requests propagate over links that have a good SNR and via high energy nodes, which have relatively low network interface queue waiting times. The effectiveness of fuzzy logic based routing is demonstrated using data throughput, control traffic overhead, average end-to-end delay and caching performance as system appraisal metrics.

5.3.1 Fuzzy Logic Based Broadcast Limiting for Stable Route Generation

Several proposals exist for broadcast limiting in ad hoc networks and have been discussed in chapter 2. The need for redundant rebroadcast curbing is essential to reserve limited resources such as bandwidth and battery energy in mobile wireless environments. Unlike existing methods the fuzzy logic based broadcast limiting scheme developed in this study does not rely on rebroadcast probabilities, neighbour knowledge, location awareness or tuning parameters. The proposed technique investigates the use of fuzzy logic to process local link and node state parameters in determining whether to terminate a broadcast flood at node level. The fuzzy metrics used are link strength ($LS$), number of hops in a route ($NH$), remaining battery energy ($NE$) and network interface queue size ($NQ$). These parameters are chosen as they reflect the channel quality and node condition thus enabling a node to determine its suitability as an intermediate traffic router for other nodes.

The fuzzy logic controller architecture has been presented in appendix A. The broadcast limiting parameters are fuzzified by mapping them into the corresponding fuzzy sets. These fuzzy sets represent each value as being Low ($L$), Medium ($M$) or High ($H$) in the interval $[0, 1]$, with the membership functions for each parameter being shown in Figures 5.10 – 5.13. These fuzzified values are then passed to a Mamdani based fuzzy
Chapter 5 Management Architecture Evaluation

inference engine where they are used to evaluate a rule set, as discussed in chapter 3, that determines whether to rebroadcast with possible rule conclusions being terminate broadcast ($TB$) or continue broadcast ($CB$).

![Link Strength Membership Function](image)

**Figure 5.10** Link Strength Membership Function

![Current Route Length Membership Function](image)

**Figure 5.11** Current Route Length Membership Function

115
Chapter 5 Management Architecture Evaluation

Shown in Figure 5.14 is a comparison of unrestrained broadcast route discovery floods to those that have been limited by fuzzy reasoning, using the 1000 meters squared network topology shown in Figure 4.14, for \( N (N:10-100) \) nodes with \( X\% \) \((X=25, 50, 75)\) of nodes generating a route request packet. It is assumed that these nodes have no prior
awareness of each other so the source node activates a route discovery broadcast. Nodes may receive several copies of a route request but will only forward the first copy it receives. The fuzzy logic controller is tuned to select good SNR links, low number of hops in the path, and high-energy nodes, with low output network interface queues. The application of the fuzzy decision making process allows a node a certain amount of selfishness as it decides whether or not to forward a packet based on its current status and the quality of the link over which it received the packet. However, the aim of the fuzzy process is to promote network longevity as low energy nodes are spared the burden of blindly forwarding control packets, high SNR links are selected so that the probability of correct data transmissions is increased, congestion is avoided as lightly loaded nodes are used and finally paths with a small number of hops are preferred as there is less probability of link breakages due to mobility as there are fewer hops to contend with. As is evident from Figure 5.14, the utilisation of fuzzy logic impedes the flood of request packets through the network. The proposed fuzzy logic broadcast limiting algorithm is implemented at a local level by individual network nodes, however these local decisions affect global network performance. This can be likened to the concept of emergent behaviour [1], which is desirable in distributed networks such as MANETs.

![Figure 5.14 Fuzzy Broadcast Limiting](image)
The objective of the fuzzy controller while reducing broadcast discovery floods is also to promote the generation of stable routes by caching fuzzy assessed route information during the route discovery process. As route cache entries remain in a node's cache for a finite amount of time it is essential that the route be of good quality so that it is still valid for subsequent reuse. As mobile nodes have limited resources they must have a finite cache capacity for cache storage. Previous work [65] has shown that nodes which have an unlimited path cache size perform worse than nodes with a finite path cache capacity as regards to routing protocol performance. For fixed cache volumes no one capacity provides the best performance for a FIFO cache entry replacement policy with a static timeout value. Using the same path cache sizes as in [65] the impact of fuzzy logic assisted routing and variable cache capacity was investigated using the DSR protocol over the 1000 meters squared network topology of Figure 4.14 for 100 nodes, with an unlimited cache capacity and fixed cache sizes of 32 and 64 entries with a FIFO cache entry replacement policy. The simulation results shown in Figures 5.15 – 5.17 illustrate the routing protocol performance for variable cache capacities and plot the percentage rebroadcast and latency reductions achieved (Figures 5.15, 5.17 respectively) as a consequence of fuzzy assisted routing and the system throughput improvement (Figure 5.16). Clearly, a cache with unlimited size exhibits the poorest performance in terms of route request overhead, packet delivery ratio and average end-to-end latency, as per Figures 5.15 and 5.16. This meagre cache functioning can be attributed to the persistence of stale routing information in the cache. For the limited cache sizes tested, neither scope provides itself as the optimal capacity for data packet throughput and reduced delay performance because the true effect of cache size is somewhat masked by the FIFO replacement policy. Such a policy has no regard for the quality of the entry it deletes; it simply removes the current oldest entry, however this route may still be valid and more stable than other more recently cached routes. In all cases, fuzzy logic assisted routing enhances protocol performance for varying cache capacities. In particular, for the fixed cache capacities there is a reduction in the number of route request packets generated, 48% for a cache of size 32 and 54% for a cache with 64 entries. The rebroadcast reduction associated with the infinite cache capacity is not as large as the fixed capacities, as stale routing information is never expired from such a cache and so an increased
amount of route requests and subsequent rebroadcast are necessary to discover fresh routes.

**Figure 5.15** Route Request Rebroadcast Reduction with fuzzy assisted routing

**Figure 5.16** Throughput Vs. Cache Capacity with and without fuzzy assisted routing
Presented in Figure 5.17 is the percentage average reduction in end-to-end delay measurements for the fuzzy broadcast limiting scheme when compared against basic DSR. As is evident, the greatest reduction is experienced when the fuzzy limiting scheme is used in conjunction with the infinite cache type. The latency reduction as a consequence of the fuzzy is attributed to the fact that the fuzzy logic system has directed the rebroadcast of route requests and consequently the subsequent caching of routes, so better quality routes persist in a fuzzy assisted cache rather than one generated as a result of basic DSR. This means that not as many route rediscoveries are necessary and so average end-to-end delay measurements are reduced for data traffic. For the infinite cache capacity the fuzzy caching of routes while improving the quality of the routes being cached will also reduce the number of routes cached. The contents of the fuzzy infinite cache are more stable and remain valid for longer than the non-fuzzy cache. This is evident from the considerable reduction in the average latency, as the fuzzy cache contents remain valid for longer more paths are readily available for routing thus reducing the need for route discovery, so data traffic arrives at the destination faster. The decrease for the fixed capacities caches is not as large, due to the FIFO replacement
policy, which forcibly expires the oldest entries, and leads to maintaining relatively fresh cache contents. The FIFO replacement policy can expire valid paths but this is necessary due to capacity limits. However, the throughput performance and route discovery overhead achieved with fixed cache capacities are better than that achieved with an infinite capacity cache, as stale routing information persistently resides in such a cache. Figures 5.15 and 5.16 further demonstrate this and confirm the need for fixed capacity caches, as the fuzzy assisted fixed capacity caches outperform the fuzzy infinite cache with regard to throughput and rebroadcast reduction.

5.3.2 Fuzzy Logic Based Cache Management

A routing protocol should select the best path available over which to route packets. When caching routes, be it next hop information or entire paths, routes should be weighted to indicate their quality. This weighting can be used to differentiate between multiple entries for a single destination. Fresh route cache contents are essential for efficient routing. Consequently, cache entries can only persist in a node's cache for finite period of time before they must be expired. The methods considered here propose the use of fuzzy logic in assigning path weights and determining dynamic route entry expiry times.

5.3.2.1 Path Weight Assignment

In assigning weight to cache entries three possible methods were considered as outlined in chapter 3 and are surmised as:

- Use the strength of the CB decision as a weighting – Fuzzy Value
- Use the difference between the strength values for the CB and TB decisions - Fuzzy Difference
- Use a utility function – Fuzzy Utility Function. The utility function is defined as: 
  \[ FUF = \alpha DS + \beta DB + \sigma DQ + \delta DH \] in equation 3.1, where \( \alpha, \beta, \sigma, \delta \) are scaling factors representing the relative importance of their associated fuzzy metric (these
are assigned the values 0.6, 0.2, 0.1, 0.1 respectively) and \( D \) is the strength difference between the \( CB \) and \( TB \) decisions.

Table 5.6 shows an example section of a node’s route cache and demonstrates the effects of using the three proposed fuzzy assignment schemes for allocating fuzzy weight values to routes extracted from the route discovery packets, using the fuzzy metrics - current number of hops \( (H) \), link SNR \( (S) \), node battery energy \( (B) \) and present network interface queue size \( (Q) \).

<table>
<thead>
<tr>
<th>#</th>
<th>Dest</th>
<th>Path</th>
<th>Fuzzy Value</th>
<th>Fuzzy Difference</th>
<th>Fuzzy Utility Function</th>
<th>S</th>
<th>B</th>
<th>Q</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>32 719</td>
<td>0.50</td>
<td>0.75</td>
<td>0.23</td>
<td>23.61</td>
<td>4332.4074</td>
<td>28 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>45 19</td>
<td>0.75</td>
<td>0.50</td>
<td>0.67</td>
<td>29.32</td>
<td>4332.4087</td>
<td>22 2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>46 19</td>
<td>0.50</td>
<td>0.60</td>
<td>0.66</td>
<td>23.31</td>
<td>4332.4085</td>
<td>25 2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>73 19</td>
<td>0.50</td>
<td>0.35</td>
<td>0.52</td>
<td>12.56</td>
<td>4332.4083</td>
<td>14 2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>98</td>
<td>98</td>
<td>0.50</td>
<td>0.50</td>
<td>0.24</td>
<td>12.30</td>
<td>4332.4028</td>
<td>29 1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>121 54</td>
<td>0.54</td>
<td>0.50</td>
<td>0.07</td>
<td>10.84</td>
<td>4332.3999</td>
<td>27 3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>54</td>
<td>461 54</td>
<td>0.50</td>
<td>0.08</td>
<td>0.44</td>
<td>23.31</td>
<td>4332.3996</td>
<td>16 3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>54</td>
<td>73 54</td>
<td>0.75</td>
<td>0.27</td>
<td>0.52</td>
<td>12.56</td>
<td>4332.4010</td>
<td>12 2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>54</td>
<td>741 54</td>
<td>0.75</td>
<td>0.75</td>
<td>0.44</td>
<td>21.93</td>
<td>4332.3994</td>
<td>20 3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>54</td>
<td>781 54</td>
<td>0.75</td>
<td>0.75</td>
<td>0.42</td>
<td>13.79</td>
<td>4332.3992</td>
<td>24 3</td>
<td></td>
</tr>
</tbody>
</table>

The column labelled **Fuzzy Value** simply uses the strength of the fuzzy \( CB \) value, as a weight however this method is not adequate in discriminating between multiple paths for a single destination. Consider entries 1,3,4 for a node with destination address 19 – the **Fuzzy Value** assignment scheme allots each path the same value so it is not possible to say which of these paths is the best. Likewise, for the **Fuzzy Difference** column entries 9,10 have the same value (also these values are the same as those in the **Fuzzy Value** as in this case the \( TB \) decision strength is 0) again making it impossible to select the best entry. Using the **Fuzzy Utility Function** scheme dissimilar weight values are attributed to paths, with higher values signifying better paths. For entries 1,3,4 the **Fuzzy Utility Function** has distinguished between them using the fuzzy metric as specified in Table 5.6. Consequently, for destination node 19 the best path in the node cache at
present is entry 2 as overall it has the highest weighting as determined by the utility function. Similarly, entries 9, 10 can also be separated using comparable reasoning. Using this analysis, entry 8 would be deemed the most apt route for selection for destination node 54.

5.3.2.2 Cache Expiration Policy

An adaptive cache expiration policy is proposed and discussed in chapter 3, which uses the following policy to modify a static timeout value ($STV$) using the weight determined by the fuzzy utility function as follows:

\[
\begin{align*}
\text{If Fuzzy Weight } \geq 0.25 & \text{ Then } Timeout = STV + (STV \times \text{Fuzzy Weight}) \\
\text{If Fuzzy Weight } < 0.25 & \text{ Then } Timeout = STV - (STV \times \text{Fuzzy Weight})
\end{align*}
\]

Shown in Figure 5.18 is a sample of adaptive cache expiry timeout values generated using the fuzzy weight policy above, for a network of $N$ nodes over the 1000 meters squared network topology shown in Figure 4.14.

Figure 5.18 Fuzzy Adaptive Cache Entry Timeout
Chapter 5 Management Architecture Evaluation

5.3.3 Scenario Simulation Results

Using a path cache with a 64 entry capacity and a FIFO replacement policy the fuzzy logic broadcast limiting algorithm was tested using DSR, with cache entries having fuzzy logic based weight assignments and dynamic expiry times, over the 1000 meters squared topology of Figure 4.14 with the number of nodes varying from 10 to 50 and random mobility rates in the region of 0m/s – 5m/s. The fuzzy based scheme was compared against DSR with a static cache expiry time of 120 seconds and the shortest path was used as the route metric for path selection. The simulation results presented in Figure 5.19 demonstrate the effects of the fuzzy limiting scheme as applied to the route discovery broadcast floods.

![Figure 5.19 Percentage Performance Improvement with Fuzzy Assisted Routing](image)

The objective of fuzzy assisted routing is to restrict the flood of route discovery packets throughout the network and promote the generation of stable routes. By limiting the rebroadcast of route discovery packets to good quality links there will be a reduction
Chapter 5 Management Architecture Evaluation

in the number of route request packets that are rebroadcast by nodes during the route discovery process. This fuzzy rebroadcast limiting method also increases the number of route requests that successfully arrive at the necessary destination nodes and also correspondingly the number of successful route replies that are returned to the route request originator, as demonstrated in Figure 5.19. These percentage increases can be attributed to the fact that the application of the fuzzy logic system in the route discovery process lends itself to the generation of stable paths allowing broadcast packets to reach the necessary destinations thus creating stable route paths over which route replies can be back propagated towards the route request originator nodes. From Figure 5.19 it is clear that the percentage rebroadcast reduction for a small number of nodes (in comparison to the network area) is not as large when the number of nodes is increased. When the number of nodes is small in relation to the network area link breakages can be more frequent and necessitate broadcast floods to re-establish routes. As the number of nodes increases the percentage gain falls off which can be expected, as there is increased neighbour interference and additional traffic flows that lead to more collisions, retransmissions and packet losses, so performance degradation can be expected but the fuzzy assisted method for route management still outperforms basic DSR, as is demonstrated in Figure 5.19.

The node caching performance for the fuzzy and non fuzzy systems is shown in Figure 5.20 with the caching performance being expressed as the ratio of successful cache hits to cache misses. A cache hit is where a cached route is successfully used and a cache miss is when a cached path is used and results in a broken link being discovered in the path. As is evident from Figure 5.20, the fuzzy adaptive cache entry timeout mechanism results in a better cache performance, as cache records are retained for an interval that is proportional to the quality of the link over which the paths are created rather than statically timing out entries. The improved route cache functioning in turn leads to increased throughput, as per Figure 5.21, and a reduction in latencies for average end-to-end delay measurements, shown in Figure 5.22. The greatest percentage average end-to-end latency reduction is again attained for lower density networks as a result of less traffic flows and neighbour interference. As the number of nodes rises there is an
increase in neighbour interference leading to more collisions and retransmissions, thus increasing average end-to-end delay measurements.

![Figure 5.20 Caching Ratio Comparison](image1)

![Figure 5.21 Throughput Comparison](image2)
Similar results were obtained using the 50m x 20m topology and the 200 and 500 meters squared network topologies. Table 5.7 illustrates the performance achieved for a single node density over these topologies. For the 50 x 20 metres squared area, performance enhancement is not as great as the other larger topologies as this area is much smaller with greater obstacle interference, so performance gains cannot be expected to be as large as the other scenarios. There is only a slight reduction of 1.98% in route request rebroadcasting, this can be attributed to the fact that in such an environment route requests may be prevented from reaching intermediate nodes due to interference and likewise any possibly route replies. The route request source node will repeatedly attempt to discover routes and persistently generate requests until the request times out. This has a knock-on effect on the number of route requests arriving at the destination nodes and correspondingly on the number of successful route replies being returned as both the

Figure 5.22 Percentage Reduction in Average End-To-End Delays Measurements With Fuzzy Assisted Routing
larger topologies of 200 and 500 sq. meters outperform the denser area. With the larger areas there are greater improvements. With route discovery flooding being reduced and a corresponding increase in the return of route replies an increase in throughput (3% - 8%) and a reduction in average end-end delay measurements (11% - 23%) is experienced in all three test scenarios.

Table 5-7 Percentage Performance Improvement Achieved with Fuzzy Assisted Routing

<table>
<thead>
<tr>
<th>Area</th>
<th>NN</th>
<th>PRR</th>
<th>PIRD</th>
<th>PIRS</th>
<th>PICP</th>
<th>PIT</th>
<th>PRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 x 20</td>
<td>10</td>
<td>1.98</td>
<td>8.21</td>
<td>10.08</td>
<td>7.36</td>
<td>2.88</td>
<td>12.21</td>
</tr>
<tr>
<td>200 x 200</td>
<td>20</td>
<td>23.30</td>
<td>96.95</td>
<td>94.81</td>
<td>45.97</td>
<td>3.46</td>
<td>11.16</td>
</tr>
<tr>
<td>500 x 500</td>
<td>30</td>
<td>15.50</td>
<td>93.44</td>
<td>90.04</td>
<td>12.01</td>
<td>8.24</td>
<td>23.82</td>
</tr>
</tbody>
</table>

**Table Key**

- **Area**: Topology size in meters, with Voronoi Obstacle mobility model and constant mobility
- **NN**: Number of nodes
- **PRR**: Percentage reduction in route request rebroadcasting
- **PIRD**: Percentage increase in number of route request packets arriving at destination nodes
- **PIRS**: Percentage increase in number of route reply packets arriving at route request originator nodes
- **PICP**: Percentage increase in caching performance
- **PIT**: Percentage increase in throughput
- **PRL**: Percentage reduction in average end-to-end latency measurements
Chapter 5 Management Architecture Evaluation

5.4 Merging Node Awareness for MANETS

The proposed merging node awareness scheme aims to furnish adjoining network nodes with copies of current network node route cache contents so as to lessen the impact that merging nodes have on a network by providing reliable route paths. To allow existing nodes to decide whether to participate in such a scheme a fuzzy logic based system is used to assess the suitability of present network nodes as appropriate next hop neighbours for merging nodes.

5.4.1 Aperiodic Cache Provisioning for Merging Network Nodes – an initial investigation

Initial investigation into this proposed method examined the effect that procuring a neighbour node cache had on overall control traffic overhead as regards route discovery and reply packets. The network topology consists initially of 10 nodes uniformly positioned over the available network area. Node traffic generation sessions are established at arbitrary times over the first 60 seconds of simulation time, for initial network nodes, so as to vary the network load and remain active for the complete simulation. Merging nodes join the existing network at random intervals over the simulation time.

The objective of the aperiodic cache provisioning for merging network nodes is to reduce route discovery/reply routing overhead by minimizing the route discovery floods necessary to build a functioning cache so that a node may undertake data transmission. As merging network nodes are prohibited from transmitting any self-generated data traffic until it receives a cache a node cannot indefinitely seek a cache and abort after \( n \) attempts. Failure to do so may lead a node to congest the network with these requests. A node may fail to receive a cache reply due to collisions or at its current location it may be isolated from the rest of the network so cache requests may be futile. In order to ascertain how many attempts a node should have for obtaining a cache, simulations were run with joining nodes being forced to wait for a cache reply before beginning transmission of their own data traffic. However, nodes are not prohibited from processing other network traffic.
traffic. Even with this restriction the cache provisioning technique results in less route discovery/reply control traffic than without as is evident from the results shown.

Initial simulations were carried out with merging nodes issuing unlimited cache requests that cease once a cache reply is received. The interval between successive cache requests was varied over 1ms, 2ms, 5ms and 10ms with 5, 15, 25 nodes being added at random intervals over the complete simulation time. The results shown in Figure 5.23 are for DSR, with and without cache provisioning and unrestricted cache requests with cache request intervals of 1ms, 2ms, 5ms, and 10ms. The results indicate that cache provisioning for merging nodes is constructive in terms of route discovery/reply control traffic reduction as with the provisioning scheme less route discovery/reply packets were generated as a consequence of the new nodes. Shown in Figure 5.24 are the same set of results with the overhead as a consequence of cache provisioning being included. Again it is evident that the cache provisioning scheme is still more beneficial even when considering the additional cache provisioning overhead.

![Figure 5.23](image)

**Figure 5.23** No Cache Provisioning Vs. Cache Provisioning for varying cache intervals with unlimited cache requests
The selection of the delay between repeated cache requests was set to 5 milliseconds as nodes should not have large waiting periods before they begin transmission of their own data traffic as this will only lead to a nodes’ buffers filling and possibly overflowing. Transmissions over the channel last typically around 600μs so an initial wait period of 1ms between continual cache requests was examined. Nodes returning any cache in response to a cache request delay their reply by a small jitter period chosen randomly between 0 and 500μs so as to prevent a cache reply storm from several nodes, thereby reducing the possibility of collisions and consequently more control traffic. Therefore, a delay of 1ms is unsuitable between successive cache requests as the round trip time of sending a cache request + jitter delay before sending a reply + a cache reply can possibly be greater than 1ms. The unsuitability of this interval of 1ms is evident from Figure 5.24 as this results in the production of most route discovery/reply traffic. Examining the route discovery/reply traffic generation for delays of 2ms, 5ms, 10ms, as per Figure 5.24, there is little to distinguish one interval from another. Shown in Figure 5.25 is the total cache provisioning overhead generated as a consequence of
adding 25 nodes to the network. Inspecting the overhead associated with the various delays, as per Figure 5.25, indicates that either 5ms or 10ms is more suited as they generate similar amounts of cache provisioning overhead. Finally, a delay of 5ms is selected, as the interval between successive cache requests, as using a delay of 10ms does not significantly lessen route discovery/reply traffic and cache provisioning overhead.

![Figure 5.25 Cache Provisioning overhead Vs. varying delay between successive cache requests](image)

Both Figures 5.23 and 5.24 have demonstrated the effectiveness of cache provisioning in reducing control traffic overhead. However, a node cannot have an arbitrary number of attempts at securing a cache as this prevents nodes from transmitting their own data traffic and drains their battery and uses bandwidth. Merging nodes accept the first cache reply they receive and ignore all other cache reply responses making the node that sends the first cache reply as the next hop for the merging node's cache records. For a cache provisioning scheme using a delay of \(5\text{ms}\) a cache request limit \(n\) was determined by examining the overhead associated with \(n\) having an integer value in the range \([1,5]\). The upper limit on this range was set to 5 as increasing beyond this would lead to excessive wait times for cache relies. It was found that having \(n = 2\) produced the least amount of overhead. Consequently, these values of \(n = 2\) and \(5\text{ms}\) were selected as simulation parameters for the cache provisioning scheme. The effectiveness of the approach is demonstrated in Figure 5.26 when compared against a scenario that employs no provisioning for merging nodes. Also, including the control overhead as a result of
Chapter 5 Management Architecture Evaluation

cache provisioning demonstrates that cache provisioning is still more effective for merging network nodes, as demonstrated in Figure 5.27.

**Figure 5.26** No Cache Provisioning Route Discovery/Reply overhead Vs. Route Discovery/Reply with Cache Provisioning

**Figure 5.27** No Cache Provisioning Route Discovery/Reply overhead Vs. Route Discovery/Reply with Cache Provisioning overhead included
5.4.2 Merging Node Cache Provision using Fuzzy Logic

As is evident from the simulation results presented, the cache provisioning scheme is beneficial for merging nodes. This scheme was modified to allow a merging node to generate a cache using the best subset of cache records from all nodes that disclosed their cache contents. In conjunction with this, the scheme was extended to allow existing network nodes to decide using fuzzy logic whether to convey their cache records. This is to enable merging nodes to possibly distribute traffic over several next hop nodes rather than an individual node and to route via nodes that are willing next hop neighbours. As a consequence of this scheme the route request overhead associated with merging nodes should be reduced when compared with a scenario that does not employ merging node awareness. Likewise, as routes are readily available to merging nodes the amount of self-generated data packets placed directly on the merging node queues should be increased. As an ancillary effect of this the average end-to-end delay performance for data traffic should be lessened as the merging nodes generate less route requests and ensuing broadcast floods, so data traffic should be pushed through the network faster.

The fuzzy metrics used are congestion status ($CS$), neighbour reachability ratio ($NRR$) and remaining node battery energy ($NE$). The fuzzy evaluation of these parameters allows a node to assess its suitability as a next hop neighbour for merging network nodes. The fuzzy logic controller is again based on the architecture that has been presented in Appendix A, with the metrics being fuzzified by mapping them into the corresponding fuzzy sets. The $CS$ fuzzy set represents each value as being Very Low ($VL$), Low ($L$), Medium Low ($ML$), Medium High ($MH$) or High ($H$) in the interval $[0, 1]$. The $NRR$ and $NE$ fuzzy sets represents each value as being Low ($L$), Medium ($M$), or High ($H$) in the interval $[0, 1]$. The membership function for the $NRR$ and $CS$ are shown in Figures 5.28 and 5.29, the $NE$ membership function used is the same as the one presented in Figure 5.12. The fuzzy logic controller associated membership functions were again manually tuned for optimal performance using extensive simulation runs.
Using a Mamdani based fuzzy inference engine the decision to disclose cache contents (DC) or not (NS) is determined. Example rules for both rule conclusions are shown below:
Chapter 5 Management Architecture Evaluation

$$\text{IF} \ (CS = H) \ \text{AND} \ (NE = H) \ \text{AND} \ (NRR = L) \ \text{THEN} \ NS$$
$$\text{IF} \ (CS = ML) \ \text{AND} \ (NE = L) \ \text{AND} \ (NRR = M) \ \text{THEN} \ DC$$

Fuzzy rules are evaluated by taking the minimum fuzzified parameter as the overall rule strength. The maximum rule strength value for both the $DC$ and $NS$ evaluations are used as the conclusion for both possible outcomes. Table 5.8 lists the fuzzy states for $CS$ and $NE$ that in conjunction with the $NRR$ metric, assuming medium or high state values, results in a $DC$ consequent rule evaluation.

<table>
<thead>
<tr>
<th></th>
<th>NE = L</th>
<th>NE = M</th>
<th>NE = H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CS = VL$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$CS = L$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$CS = ML$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$CS = MH$</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$CS = H$</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The merging node awareness algorithm was tested using DSR over the 1000 meters squared topology of Figure 4.14 using an initial node density of 15 nodes with 10 merging nodes joining the network after 30 seconds at random intervals during the simulation as shown in Figure 5.30. The simulation results are presented in Figure 5.31 and show the percentage performance enhancement for merging nodes that have cache records made available when integrating with an existing network. Two schemes have been implemented and are compared with a scenario that does not furnish cache records. The $All \ Nodes$ method forces all neighbouring nodes of the merging node to make available their cache whereas the $Fuzzy \ Nodes$ scheme enables neighbouring nodes to decide using fuzzy logic whether to impart their caches. For both methods the merging nodes construct a cache from the best available subset of cache records. As is evident from Figure 5.31, the fuzzy logic based scheme $Fuzzy \ Nodes$ results in the best network performance for merging nodes as this method gives rise to:
Chapter 5 Management Architecture Evaluation

- The best percentage decrease in route request overhead generated by merging nodes
- The highest percentage increase in the amount of self-generated data traffic being directly placed on merging node queues
- The greatest reduction in average end-to-end delay measurement for data traffic through the network

With the *Fuzzy Nodes* method there are less nodes providing cache records but those nodes that are doing so have gauged themselves as appropriate next hop neighbours for the merging node.

![Figure 5.30 Merging Node Simulation Join Time](image)

![Figure 5.31 Merging Node Awareness Performance Measure](image)
5.5 Discussion

This chapter has presented an evaluation of the proposed dynamic route management strategies for mobile ad hoc networks. Fuzzy logic and policy based methods are used to introduce an adaptive architecture for application with MANET routing protocols with a view to enhancing network performance.

Policy management, usually centrally operated, has been used in a distributed manner to implement a congestion control scheme for distributed networks, such as MANETs. Nodes that are likely to become congested or are currently congested seek to assuage their congestion status by notifying their one-hop neighbours of their congestion status. This prompts the local neighbourhood to redirect traffic if possible, away from the congested area and to dynamically reroute traffic via non-congested nodes so that collisions and retransmission are eased. An evaluation of the proposed method verified its effectiveness as the average queue and transmission ratio statistics were improved and the average end-to-end delay measurements were reduced.

Fuzzy logic is used to develop a rebroadcast flood limiting algorithm to curtail non-optimal network floods that assesses both node and channel quality in making the rebroadcast decision. The strength of the fuzzy rebroadcast YES decision is used to weigh cache records to indicate quality and to differentiate multiple cache entries for a single destination. This strength value is also used to implement a dynamic cache entry expiry policy to adaptively timeout records. An analysis of the proposed system has demonstrated improved network performance with increases in throughput, caching action, route requests arriving at the necessary destination nodes and route replies being received by route request originator nodes while there is a reduction in route request rebroadcasts and average end-to-end delay measurements.

For integrating new nodes into an existing ad hoc network, the concept of cache provisioning was proposed. This method aims to furnish merging nodes with active routing information so that the initial route request bursts associated with nodes having no network topology knowledge are reduced. By providing route records merging nodes can establish an immediate cache and so can place an increased number of data packets directly on their network interface queue without the need for route discovery, which also
Chapter 5 Management Architecture Evaluation

reduces average end-to-end delay time for data traffic. The simulation evaluation presented confirms these premises.

The proposed overlay route management architecture endorses distributed decision making by locally managing nodes while enhancing global network performance. To gauge the performance of any proposed strategies the cost of implementing/tuning these methods must be such that it does not outweigh the beneficial effects of these strategies. For the overlay architecture presented in this thesis a set of fuzzy membership functions, fuzzy rules and policy statements are used to implement the proposed route management strategies. These rule sets, policies and membership functions required intensive simulation investigation to tune the system parameters, this lead to a set of generic parameters that are suitable for general environments. Such extensive off line tuning may not be cost effective in real world environments and could outweigh the beneficial effects of this and indeed any other performance enhancing techniques. As ad hoc network deployments are not readily available in real world environments it is difficult to examine the cost of implementing such performance enhancing strategies. However, this is a factor that must be considered with real world deployments with due consideration being given to computational overhead introduced and hardware requirements as a consequence of performance enhancing techniques.
Chapter 6 Conclusions & Future Work

The study presented in this thesis investigated the application of a management framework that acts as an overlay architecture for mobile ad hoc network routing. With advancing wireless network technologies mobile computing is expected to become increasingly widespread. To furnish this vision, mobile ad hoc networks must be robust and provide efficient operation by integrating routing functionality into mobile computers. Mobile computers must cope with dynamic, possibly rapidly varying, multihop topologies that comprise of low power and energy limited devices, which communicate over bandwidth constrained wireless links.

Infrastructured networks rely on Mobile IP to support roaming with routing being implemented via home and foreign agents. On the contrary, the aim of ad hoc networking is to autonomously manage mobility with the present network nodes forming the network infrastructure and providing routing functionality in an ad hoc manner without the support of centralised control. Ad hoc network clusters can operate autonomously and effectively manage mobility however, such clusters are likely to need a gateway to the fixed Internet and so hosts must be provided with or themselves provide roaming capabilities over diverse network technologies.

The need for mobile computing has driven the research in the field of ad hoc networks and was used as the motivation for the work presented in this thesis. The main contribution of this work is the development for a dynamic route management architecture, which can be incorporated with existing MANET routing protocols, that utilises current network conditions to direct network routing operation and enhances network performance in terms of increasing throughput while reducing network delays and control traffic overhead. The network management architecture, discussed in detail in chapter 3, consists of three main elements namely broadcast limiting with fuzzy caching, congestion monitoring and merging node awareness. The following section summarises the main contributions and conclusions that can be drawn from the work presented in this
thesis and the chapter concludes with indications on how to extend the presented management framework.

6.1 Discussion

The results presented in this thesis have compounded the need for network and node self-awareness consideration when implementing routing protocol strategies. The proposed overlay route management architecture is flexible and can be adapted for use with current MANET routing protocols. Several routing protocols for MANET environments have been proposed, each protocol exhibiting varying performance characteristics over sample network environments. Ad hoc network configurations can differ greatly depending upon the topology, traffic flows, mobility rates and node densities consequently, some protocols can outperform others depending on the particular network scenario. As ad hoc networks exhibit time varying characteristics routing protocol efficiency can have an irregular performance profile. While it may be feasible to dynamically tune some routing protocol parameters to improve performance the underlying routing protocol functionality remains unchanged. As the routing protocol has no control over network conditions, it is the individual networks nodes or clusters of nodes that must react to present network conditions.

The proposed overlay route management architecture enables nodes to perceive their state and the local environment within which they are situated, and their reactions can be directed based upon these observations. Individual node behaviour is inextricably correlated with that of its adjacent neighbours. Nodes while concerned with maximising their performance contribute to enhancing global network performance by locally managing their network space. The proposed overlay route management architecture promotes distributed decision making while enhancing global network performance. The overlay architecture presented could be incorporated with little overhead in terms of hardware as digital logic. can be used to implement the fuzzy logic and policy decision making strategies. The proposed route management methods can be implemented as a complete architecture or as individual route management strategies. However, as the proposed route management methods complement each other the ideal scenario is to have
all network nodes implementing the complete overlay architecture. The merging node awareness scheme is effective only when nodes join a network whereas as the fuzzy logic assisted routing and congestion monitoring strategies can have beneficial effects throughout the network lifetime. The fuzzy assisted routing addresses broadcasting limiting in the route discovery process and the weighting of route cache entries for an adaptive expiry policy. For maximum effect these strategies should be implemented as a unit. The fuzzy broadcast limiting generates a subset of high quality routes using multiple metrics and these same metrics are then used to weigh the individual routes created and consequently generate adaptive timeouts. The adaptive timeouts effectively give the fuzzy generated routes a precedence order with the better quality routes persisting longer in a node’s cache. Having generated a subset of the best available routes with the use of fuzzy routing, this can conceivably lead to a scenario where individual nodes are involved in many routes and can depending on network traffic flows become congested. The application of the congestion monitoring scheme is then necessary. The proposed overlay architecture is a software solution that looks towards improving network throughput, reducing control traffic overhead and minimising average packet latencies. For delay sensitive applications like voice over IP and video on demand, ad hoc networks that can support QoS level agreements as regards delay and jitter will drive the potential for using such networks in real world environments.

A current strain of interest for communications network research is the concept of autonomic computing [130]. This is a term that was coined by IBM in 2001 to describe computing systems that are said to be self-managing. As networks are becoming increasingly complex, a desirable trait is that they can self-configure and adapt to changing network conditions in terms of topology, traffic flows, connectivity and such like. For a system to support autonomy, network strategies must be implemented in a distributed manner, be capable of observing changes in the network and adapting to the current conditions. The proposed route management architecture can be viewed as an initial step towards developing an autonomic route management framework that’s aims to furnish nodes with a self-management autonomic intelligence that augments node routing functionality. As MANET environments are inherently distributed systems the route management architecture presented in this thesis does not make use of system-wide
knowledge and relies on individual nodes having only local network perceptions. The proposed overlay architecture looks towards individual nodes achieving self-management through an information infrastructure that generates system knowledge, consisting of both self and local channel awareness. Using this system knowledge decisions are inferred and the decisions taken are then used to modify a node’s routing capability and routing protocol parameters.

For MANET evaluations it is common with most research that MANETs are assessed over closed simulated environments. Uniform network conditions are experienced over the available simulation space, with node density, node mobility rates and traffic flows remaining constant for the simulation duration. Such an approach does allow for the testing, evaluation and comparison of routing protocols under the same conditions but it is not an accurate reflection of realistic environments over which ad hoc would be deployed. Real world scenarios will not conform to a uniform space with similar conditions being experienced throughout the operating area. For future ad hoc networks to be successfully deployed in real world environments they must operate with an efficient distributed management architecture that supports auto-reconfiguration as demanded by the network conditions, have the ability to utilise several air interface, be relatively long-lived and be capable of supporting heterogeneity among devices.

Diverse operating conditions will drive the need for an autonomic reconfigurable network layer that will facilitate node adaptation to current networking conditions. In conjunction with supporting multiple air interfaces, devices and subsequently the network layer must be capable of providing and influencing the selection of the most suitable routing protocol to satisfy the prevailing network state.

Both the National Institute of Standards and Technology, NIST, [131] and the MANET working group [132] have suggested the metrics, control overhead, end-to-end delay and throughput as being suitable evaluation metrics for quantifying the performance of a routing protocol. However, other metrics such as energy expenditure, path length, and queuing performance have also been employed. As discussed in chapter 3 there is an urgent necessity to define a benchmark suite of tests for routing protocol performance evaluation. In conjunction with this the metrics suggested by NIST and the MANET working group should be adopted as the standard set of performance appraisal
Chapter 6 Conclusions & Future Work

metrics. Such a combination of standardised benchmark testing and metrics will lead to a coherent research effort with quantifiable evaluation results.

6.2 Review of Contributions

As ad hoc networks operate in a decentralised way, self-organising behaviour is essential in implementing a management architecture. For the overlay management framework developed as part of this work, node adaptability has been realised through the interpretation of self and local channel awareness. Using this system knowledge dynamic route management strategies have been accomplished. In the following the main contributions of this study are outlined and the main conclusions that can be drawn from the results are presented.

6.2.1 Congestion Monitoring

The proposed Congestion Monitoring approach is a reactive method for congestion control that uses a distributed policy technique to enable nodes to cyclically determine their congestion status. Nodes that are approaching congestion or are congested attempt to alleviate this congestion by issuing broadcast warnings and notifications to their immediate neighbourhood. The local neighbour reaction is to attempt to divert traffic away from congested network zones and to adaptively redistribute traffic over lightly loaded network areas to reduce network collisions and retransmissions. The effectiveness of this method was evaluated using queue ratio, transmission ratio and reduction in average in average end-to-end delay measurements as performance metrics. From the simulation results presented in chapter 5, it is clear that addition of this distributed congestion control management strategy is beneficial as the average queue ratio and transmission ratio statistics are lowered which is indicative of an improvement in network interface queue performance and reduction in the numbers of retransmissions. As a consequence of this, the average end-to-end network delay is also greatly improved.
Fuzzy logic has been used to develop a rebroadcast strategy to curb non-optimal floods associated with the route discovery process and has been applied to the DSR protocol to illustrate the effectiveness of the proposed scheme. The fuzzy logic system employs link strength, node energy, current network interface queue size, and number of hops currently in the route path as fuzzy metrics. Multiple metrics were used to make the routing protocol more robust rather than simply optimising a single parameter in the routing decision. An individual node decides using the fuzzy system to rebroadcast route discovery packets based on the fuzzy decision. The fuzzy logic system is used to develop an adaptive behaviour as a node interprets its own state, based on its remaining battery life and network interface queue size, while also assessing the viability of possibly generating a route path via itself based on the link strength and current path length. The fuzzy system limits rebroadcast floods to the best quality paths in terms of hop count, strong links and high-energy lightly loaded nodes so that high quality paths are generated for traffic routing. Nodes that decide to continue the flooding process cache the route information extracted from the route discovery packet header.

As nodes can have multiple paths for a single destination the strength of the fuzzy decision was used to assign a weight to cache entries for path differentiation. To purge node caches of stale routing information the strength of the fuzzy decision was also used to adaptively modify a static route entry timeout value so that cache contents are expired based on the quality of the path that generated the cache record.

The simulation results presented in chapter 5 for the fuzzy assisted routing demonstrate the effectiveness of the proposed management scheme. Application of the fuzzy logic based route request rebroadcast method limits the spread of route request packets being flooded through the network while also increasing the number of route requests that arrive at the necessary destination nodes and increasing the quantity of route replies that are successfully returned to the route request originator. The beneficial effects of this method over the test scenarios are demonstrated through significant reductions in average end-to-end latency measurements, with the need for route request rebroadcasting being limited while the number of successful route replies being returned is increased.
Chapter 6 Conclusions & Future Work

These corresponds to an increase in caching performance and consequently in data throughput.

6.2.3 Merging Node Awareness

The concept of merging node awareness was proposed as a method of integrating merging nodes with an existing network. This involves current network nodes making available a copy of their route cache contents to merging nodes, with a view to lessening the impacts of these merging network nodes by possibly reducing control traffic bursts associated with these nodes as they attempt to construct functioning route caches. A preliminary investigation into developing this hypothesis looked at the efficacy of this method from the merging node perspective and was reasoned to be advantageous for merging nodes as the control traffic route request overhead was lessened when route cache records were provided to merging nodes. This premise was then extended to allow existing nodes to decide using fuzzy logic whether to impart cache contents. The intention of this is to allow existing nodes to assess their suitability as possible next hop nodes for traffic generated or forwarded by a merging node. Nodes use congestion status, neighbour reachability ratio and remaining battery energy as fuzzy metrics in this decision making process. Merging nodes build an initial route cache using the subset of best paths from all the cache contents it receives. The simulation results presented for this scheme show that the overall route request overhead is lessened, average end-to-end delay is reduced and merging nodes as a consequence generate less route request overhead.

6.3 Future Work

The results of the study presented in this thesis have shown the beneficial effects that incorporating an overlay management framework with a standard MANET routing protocol has on system performance. MANET mobility patterns are derived based on synthetic models. For routing protocol evaluation it is critical that the mobility model mimics realistic movement as routing protocol performance is strongly correlated with
Chapter 6 Conclusions & Future Work

the accuracy of the mobility model. Accurate mobility modelling is critical in evaluating ad hoc network performance and the obstacle mobility model used in this work goes towards imitating realistic movement patterns by incorporating movement around and via obstacle using Voronoi polygons. However, this model like the random waypoint model has nodes moving between random start and end points over the available topology space. For more effective and realistic mobility modelling the determination of a new movement point should be based on a prior movement history that records previously visited destinations, velocities and directions. Extending the Voronoi Obstacle mobility model to select the next destination based on previously visited destinations will produce more realistic movement traces. Consider the movement of people, they do not move between arbitrarily connected points at random and do not repeatedly visit the same point.

The distributed reactive policy based congestion control scheme presented in this study relies on nodes making local decisions as they operate in a decentralised manner. Congestion is local in networks, however if several nodes within a local neighbour are congested then a scheme such as the one presented could suffer from a ping-pong effect where congested neighbours attempt to alleviate their congestion by redistributing traffic away for themselves and towards their neighbours. A node may temporarily assuage its congestion state but as it resides in a congested network zone it will resort again to being congested and may continue to alternate between states. To discourage such behaviour nodes should be aware of the congestion state of its neighbours and of its own congestion performance and use this as an indicator to perform congestion control or not. This action can be likened to adopting swarm behaviour in local decision making and could enhance the performance of the proposed congestion control scheme.

As ad hoc networks are time varying systems their performance could conceivably be improved by using adaptive methods to implement management techniques. The proposed fuzzy logic based routing strategies could be enhanced by developing an intelligent hybrid system that uses evolutionary learning techniques, such as neural networks or genetic algorithms, to tune the membership functions of the fuzzy system. Though fuzzy logic can encode expert knowledge directly using rules with linguistic labels, it can take an appreciable amount of time to design and tune the membership functions, which quantitatively delineate these linguistic labels.
Chapter 6 Conclusions & Future Work

Evolutionary algorithms could be employed to automate the tuning process, using learning algorithms and training data to develop expert knowledge from system data.
A.1 Fuzzy Logic Set Theory

Lotfi Zadeh introduced fuzzy set theory, which forms the basis of fuzzy logic, in 1965 [73]. Zadeh stated that the true/false nature of boolean logic did not consider the varying shades of grey or uncertainty found in the real world. To represent these levels of greyness between the extremities of true and false, Zadeh proposed fuzzy sets. Fuzzy sets consider degrees of membership rather than stating that an element is 100% a member of one set only. With fuzzy sets an element can be a partial member of multiple sets. Fuzzy logic is used in a decision making process that aims to interpret human expertise and deal with uncertainty. Fuzzy logic theory applications are typically predisposed towards engineering problems. However, fuzzy reasoning has also been successfully employed in areas such as psychology, medical diagnostics, management, etc. Fuzzy logic facilitates the modelling of imprecision and uncertainty expressed by human observations in the decision making process. Fuzzy logic does not use intransigent borders for decision evaluation instead it uses overlapping limits to subsume varying levels of greyness in the decision making process.

A.2 Fuzzy Logic Reasoning System

The principle of a fuzzy logic reasoning system (FLRS), as shown in Figure A.1, is to use human expert knowledge to define a rule set that allows the extraction of decisions or actions that are considered to emulate human reasoning [73, 74]. A FLRS receives a priori qualitative data as system input. The elements of a FLRS are

- Input data fuzzification
- Rule Evaluation
- Composition
Appendix A Network Management Tools

- Defuzzification

Fuzzification describes the fuzzy membership function sets and the determination of the degree of membership of the crisp input data of the corresponding membership function set. Rule evaluation resolves the firing strength of each rule in the fuzzy rule set. Composition is the combination of the fuzzy rule set output evaluations. Defuzzification is the translation of the fuzzy output to a crisp output value.

![Diagram of Fuzzy Logic Reasoning System](image)

Figure A.1 Fuzzy Logic Reasoning System

### A.2.1 Fuzzy Inference

Fuzzy inference is typically performed using either Mamdani or Sugeno fuzzy inference [73, 133]. The purpose of an inference engine is to perform fuzzification and rule evaluation. Fuzzification takes the crisp input values and represents them in terms of the fuzzy set membership functions, which are defined over the range of the fuzzy input values and linguistically describe the crisp input variable's universe of discourse. Rule evaluation applies the fuzzy rule set to the fuzzy inputs generated by the fuzzification procedure with each rule being resolved separately. The fuzzy rule set is formed using conditional IF-THEN statements that describe the action to be taken in response to the
Appendix A Network Management Tools

fuzzy inputs, as per Figure A.2. A fuzzy rule consists of a fuzzy input and the consequent is the rule output. Mamdani and Sugeno inference are similar with the only difference being that Sugeno uses a mathematical function instead whereas Mamdani inference uses a fuzzy set in the rule consequent. Mamdani style inference has been considered in the application of a FLSR for MANET route management as this method is widely accepted for capturing expert knowledge and allows the expertise to be described in a more intuitive manner than the Sugeno method, which is said to lose linguistic interpretability. As only Mamdani inference has been considered in this work Sugeno inference is only briefly described.

![Figure A.2 Fuzzy Rule](image)

A.2.1.1 Mamdani Fuzzy Inference

A.2.1.1.1 Fuzzification

Fuzzification transforms crisp inputs to fuzzy inputs using membership functions. Fuzzy labels, such as LOW, MEDIUM or HIGH, are assigned to the universe of discourse of each crisp input value [73, 74, 133]. The membership function defines the range of input values that correspond to the fuzzy labels. Input values can be represented by more than one label, as per Figure A.3, as there is an overlap between label boundaries allowing for the gradual change from one label set to the next. An input membership function set \( \mu \) is generated by determining the degree of membership (value between
Appendix A Network Management Tools

0...1) of the crisp input value \( X \) of the membership function labels as shown in Figure A.4.

### A.2.1.1.2 Rule Evaluation

The input membership function set \( \mu \) is applied to the antecedents of the fuzzy rules [73, 74, 133]. For fuzzy rules involving multiple antecedents the fuzzy operator AND or OR is used to determine a single output number that represents the rule evaluation or firing strength of a rule. The OR union operator is used in the disjunction of the fuzzy rule antecedents with the maximum degree of membership value being selected as the rule evaluation.

\[
\mu_A \lor B(x) = \max[\mu_A(x), \mu_B(x)]
\]  

![Figure A.3 Logic Membership Functions](image)

![Figure A.4 Degree of Membership Evaluation](image)

\[\mu(x = \text{LOW}) = 0\]
\[\mu(x = \text{MEDIUM}) = 0.4\]
\[\mu(x = \text{HIGH}) = 0.6\]
Appendix A Network Management Tools

The AND intersection operator is used in the conjunction of the fuzzy rule antecedents with the rule evaluation being selected as the minimum of the degree of membership values.

\[ \mu_A \land B(x) = \min[\mu_A(x), \mu_B(x)] \]

Consider a two-input one-output system:

**Inputs:** \( \mu_{x=HIGH} = 0.30 \) \& \( \mu_{y=LOW} = 0.55 \)

**OR operator:** IF x is HIGH (0.30) OR y is LOW (0.55) THEN z is 0.55

**AND operator:** IF x is HIGH (0.30) AND y is LOW (0.55) THEN z is 0.30

A.2.1.2 Sugeno Fuzzy Inference

Sugeno and Mamdani fuzzy inference methods are similar [73, 133]. Mamdani inference uses a fuzzy set for the rule consequent whereas Sugeno inference uses a mathematical function of the input variables. A Sugeno fuzzy rule is of the form:

IF \( X \) IS A AND y is B THEN \( z = h(x, y) \)

Where \( A, B \) are fuzzy sets in the antecedents and \( z = f(x, y) \) is a polynomial crisp function in the consequent. The system output is a weighted average of the rule evaluations.

A.2.2 Composition

For fuzzy inference there are two general methods for applying the result of the antecedent evaluation to the membership function of the consequent. The Clipping or Alpha-Cut method is the most common method used as a means of aggregating the fuzzy rule evaluations [73]. The output membership function of the consequent is truncated at the antecedent degree of truth or membership level as shown in Figure A.5.

![Figure A.5 Clipping](image-url)
Appendix A Network Management Tools

With the scaling method the original consequent membership function is adjusted by multiplying all its membership degrees by the truth-value of the rule antecedent. The scaling method offers a better approach for preserving the original shape of the fuzzy set however, the clipping method is preferred as it involves less complex and faster computations and produces an aggregated output surface that is easier to defuzzify. Composition is the aggregation of the set of clipped or scaled rule outputs into a single fuzzy set.

**A.2.3 Defuzzification**

Several defuzzification procedures [73, 133] exist but the most commonly used are the Centre of Gravity (COG) method and the Maximum method. In theory the COG should be calculated over a continuum of points in the output universe of discourse, however, a reasonable estimate can be calculated over a sample of points in the output range domain, as per equation A.3.

\[
COG = \frac{\int_{a}^{b} \mu_{A}(x) \, dx}{\int_{a}^{b} \mu_{A}(x) \, dx}
\]  

A.3

For the Maximum method, the variable value at which the fuzzy subset has its maximum truth-value is chosen as the crisp value for the output variable. There are several variations of the Maximum method that differ in the way they manage when there is more than one variable value at which this maximum truth-value occurs. One of these is the Average-of-Maxima method that returns the average of the variable values at which the maximum truth-value occurs.

**A.2.4 Tuning & System Enhancement**

Tuning the fuzzy system is done by changing the rule antecedents or consequents, changing the cross points of the system membership functions, or adding additional degrees to the membership functions. The inclusion of new levels would generate
Appendix A Network Management Tools

additional rules. Tuning can be done manually through the use of system performance evaluation via simulation or by using learning techniques, such as genetic algorithms or neural networks, to adaptively tune rule sets and membership functions.

A.3 Policy Based Networking

In policy based networking, a policy is a formal set of statements that define how a network's resources are to be allocated among the network clients [75]. Network clients can be individual users, host computers, or applications. Network resources can be apportioned based on factors such as client or application priorities, available resources and current network load. Policy resource distribution can be either static or dynamic. Policy based networking has been applied to a variety of networking applications, such as call admission traffic management, congestion control and QoS guarantying.

![Policy Engine Diagram](image)

**Figure A.6 Policy Engine**

Policy system architectures are described based on the relationship between the points where the outcome of a policy is enforced, the Policy Enforcement Point (PEP) and the point where the policy decision is evaluated, the Policy Decision Point (PDP), as shown in Figure. A.6. The fundamental building block of a policy is a policy rule, which is a declarative statement that relates a policy object with a value. For example, a policy rule can define an action such as Application Priority = HIGH or a condition such as
Appendix A Network Management Tools

Traffic Load = Heavy. Policy rules are written to describe either conditions or actions, with rules including possibly several conditions and actions. For policy based networking policies are stored centrally and can shared among several network management applications. Policy management rules are stored in a policy repository.
Appendix B MANET Routing Protocols

B.1 Unicast Topology-Based Routing Protocols

Typically, MANET communication is based on unicast routing [2, 112]. Such routing protocols depend on routing tables or caches that indicate the next hop that a packet is to be sent over on as it travels towards the necessary destination. If a network node is unaware of the next hop then, through a route discovery process it generates a route. As MANET nodes can roam freely the network topology is arbitrary and can change frequently. To counteract the mobility effect unicast protocols depend on routing tables and caches to provide fresh packet routing information in the form of the packet next hop address or provision of the complete source-destination path next hop. Routing tables and caches are produced and maintained in dissimilar manners and this is what separates unicast protocols into topology-based or geographical-based protocols.

B.1.1 Proactive Routing

Proactive or table driven unicast routing involves each node maintains a table of all possible paths to every node within a network. All nodes update these tables keeping the network status current and if a change in topology occurs update information is propagated through the network and nodes update their tables accordingly [2, 112]. When a node needs to communicate it simply refers to the table for the destination route. Table driven based protocols consume a major part of the transmission bandwidth and battery power in maintaining these routing tables.

B.1.1.1 Destination Sequenced Distance Vector - DSDV

DSDV is based on the Bellman Ford (Distance Vector) algorithm. Each node advertises to its current neighbours its own routing table; which includes all possible
Appendix B MANET Routing Protocols

destinations, the number of hops to them (distance metric), next hop node, and the (even) sequence number (originated by the destination node) assigned to each path [20]. Each node periodically broadcasts a monotonically increasing even sequence number. These numbers are used to distinguish stale routes from current ones and thus avoid loop formation. If two nodes possess the same sequence number then the route with the smallest distance metric is used and a route with a greater sequence number is preferred as it is considered as being the most recent.

Link failure can occur due to node mobility. A broken link is assigned a distance metric of $\infty$ (any value greater than the maximum allowed distance metric). When a node detects that a link to the next hop has failed, any route through that hop node is assigned an infinite metric and an odd sequence number (even sequence number incremented by 1) by the source node and is advertised immediately [112]. If a finite path is found after a link failure event this path is now assigned an even sequence number and a finite metric. This will trigger an update and this information is diffused through the network.

The DSDV protocol is based on periodic updates – in order to reduce the amount of update information and bandwidth use, two types of information exchange are employed: Full Dump or an Incremental Update. A full dump sends the entire routing table to a node’s neighbours and could encompass several network data packets. In contrast, an incremental update will only carry information about those routing table entries that have had a distance metric change since the previous update and this data must fit in a single packet.

Nodes must decide on how often to broadcast information. In a relatively stable network incremental updates are sent frequently in comparison to full dumps, to reduce network traffic, as there is little change in routing paths. However, in a fast changing network incremental packet size can approach that of a full dump making full dumps more numerous. When a node receives route information it must decide if this is significant enough to be part of an incremental update. If the information that a node receives is about a known route that simply has a new sequence number but the same distance metric as previous then this is considered as being insignificant and can be relayed through the next full dump. However, if an existing path exhibits a new metric then this is a noteworthy event and can be advertised through the incremental update.
information. When a node receives information on a new route it delays transmission for a finite amount of time – defined as the route settling time (based on past history the nodes estimate the settling time). This delay allows the smallest metric for a particular route to be discovered – this is termed Damping Fluctuation in the DSDV protocol [20].

DSDV is a table driven modified version of distance vector routing protocol. Every node periodically radiates updated routing information throughout the network. The loop formation problem, associated with distance vector, is prevented by attaching sequence numbers to routes with these sequence numbers being used to distinguish stale paths from current ones.

B.1.1.2 Optimized Link State Routing – OLSR

OLSR is an enhancement of the pure link state routing protocol. Firstly, it compresses the size of the update information sent in control packets by using a subset of its neighbours, Multipoint Relay Selectors, rather then every link neighbour. Secondly, it reduces the amount of update messages that are flooded through the network by using only selected nodes, called Multipoint Relays (MPRs), to diffuse route data throughout the network [134, 135]. These optimisation methods lessen the number of retransmissions in the flooding process. Periodic HELLO messages are broadcast that contain information on a node’s neighbours and their link status with that node. Topology Control (TC) packets are periodically transmitted to distribute topology information. At each node the following tables are maintained:

- Neighbour Table – Information on the neighbourhood of each node
- Topology Table – Topology data received from TC messages
- Routing Table – Routes to known destinations

For fast moving networks the interval between the periodic transmissions of TC information can be reduced so that the reaction to topological changes is optimised. OLSR maintains routes for all network destinations and thus is particularly suited to large dense networks where sizeable subsets of nodes are communicating. The denser the network the greater the optimisation achieved in comparison to the standard link state protocol. Unfailing transmission of data updates is not essential due to the periodic
relaying of control packets thus, the network can endure a loss of control data occasionally. Also, control messages do not need to be received in the order, in which they were sent, as each message contains a sequence number of most recent information [134].

The multipoint relaying concept is used to diminish the number of repeat retransmissions while propagating a TC message. This transmission reduction is implemented by a Node $N$ choosing a small subset of neighbours from its one hop symmetric neighbours (the symmetric neighbourhood of any node is the set of nodes that have at least one symmetric link [a bi-directional link between any two nodes] to that node) that envelops all nodes that are two hops away. This group of preferred neighbours is labelled the multipoint relays (MPRs) of that node. This action reduces the amount of transmissions but conveys the message to all network nodes. The smaller the MPR set the more optimal the routing algorithm is. The neighbours of a node $N$ that are not members of the MPR set, $\text{MPR}(N)$, receive and process control messages but do not retransmit these messages. Each node keeps a record of the nodes that have elected it as a MPR. This is known as the Multipoint Relay Selector Set of that node. A node acquires selector set information from periodic HELLO messages received from its neighbours. This set may alter with time and this change is indicated by the selector nodes in their HELLO messages. Each node has a specific Multipoint relay Selector Sequence Number (MSSN) associated with the MPR set. Every time this set is updated the node also augments its MSSN.

Neighbour sensing is achieved through use of the HELLO messages. Each node must detect the neighbour nodes with which it has direct and bi-directional links. The link status information is determined using the HELLO messages, these are received by all one hop neighbours but are not imparted to nodes beyond a single hop. A HELLO message contains:

- The catalogue of addresses of the neighbours with which there exists a symmetric link
- The list of addresses corresponding to heard nodes – nodes that are heard by this node (a HELLO has been received, an Asymmetric Link) but the link has not yet been determined to be bi-directional
Appendix B MANET Routing Protocols

- **MPR** – there exists a symmetric link and the transmitting node has selected this neighbour as a member of its MPR set.

Upon receiving a HELLO message, the receiving node updates its neighbour table. If the transmitting node is asymmetric in the receiving node’s neighbour table and this node finds itself in the HELLO message it upgrades the status of the link assigned to the neighbour to symmetric. If the receiving node has MPR status in the message it updates its MPR selector set appropriately. The HELLO messages allow each node to uncover its neighbours up to a distance of two hops. As a result of this information, each node selects its MPRs, which are used to flood the TC messages. Each node can build its MPR selector set using the nodes that have selected it as a multipoint relay.

Nodes use the MPR set to flood TC messages throughout the network. These messages notify other nodes of the status and any topology changes so they have sufficient information to build routes to all other nodes. A TC message includes the address of the transmitting node and a list of its MPR selector set. When a node receives a TC message it records the topology information in a topology table and based on this data each node determines the shortest path to all other nodes. If the receiving node is a member of MPR set then it retransmits the TC message otherwise it is dropped.

The addition of techniques such as Jitter and Piggybacking have been shown to improve the performance of OLSR [135, 136]. Using predetermined time intervals between the transmission of control messages (both TC and HELLO) data packets can be lost as a result of collisions. The interval time is altered by implementing Jitter – add a random quantity of time, \( \alpha \), to the control message interval, \( I \), and broadcast after \( I + \alpha \) seconds. Simulations involving the application of jitter have exhibited a significant decrease in the number of dropped packets. Piggybacking control messages is realised by delaying incoming control messages that are to be retransmitted for up to a predetermined amount of time, holdback time, before retransmitting them. Research has shown [136] when piggybacking is employed with jitter packet loss due to collisions is lessened.

OLSR is a proactive protocol, based on the link state algorithm, which employs multipoint relaying methods to reduce duplicate re-transmissions in the flooding process. It does not spawn extra control traffic in response to link failures and thus is suitable to dense networks that have high node mobility and topological variance.
B.1.1.3 Topology Broadcast Based on Reverse-Path Forwarding – TBRPF

TBRPF is a proactive, link state routing protocol that provides hop-by-hop routing along minimum hop paths to each destination [137]. This protocol employs the concept of reverse path forward (RPF) to broadcast link state updates in the reverse direction along the spanning tree (contains all nodes) produced by the minimum hop paths from all nodes to the source of the updates. The topology information gathered along the trees is used to determine the minimum hop paths that are used in the trees. To minimise overhead, each node reports only part of its source tree to its neighbours. TBRPF uses a blend of periodic and differential updates so that all neighbours are informed of the reportable part of its source tree. Also, each node can report further topology information (up to the full topology), to give enhanced robustness in vastly mobile networks. TBRPF achieves neighbour discovery using differential HELLO messages that merely report changes in neighbour status. As a consequence of this, HELLO messages are much smaller than those of other link state routing protocols such as OLSR.

TBRPF consists of two core elements: Neighbour Discovery and Topology Discovery / Route Computation. Neighbour discovery allows each node to establish with which nodes it has a direct bi-directional link and also to sense link failure. TBRPF uses Differential HELLO messages that include the list of all neighbours. These messages are broadcast at set interval times, with a small jitter. Each HELLO message contains three fields: Neighbour Request, Neighbour Reply, and Neighbour Lost. In addition each HELLO message includes the present HELLO sequence number, HSEQ that is augmented on each transmission of a HELLO message. Every node maintains a neighbour table that holds the current status of other nodes. The status of a node can be 1-Way, 2-Way or Lost. When node, say $i$, changes the state of a neighbour node, say $j$, it sends the appropriate message (Neighbour Request/Up/Lost) in at most $NBR\_HOLD\_COUNT$ (in general 3) successive HELLOs. This guarantees that node $j$ will either get the message, or it will miss $NBR\_HOLD\_COUNT$ HELLOs and as a result pronounce node $i$ to be Lost.

For topology discovery/route computation every node running TBRPF determines a source tree, which gives the minimum-hop paths to all accessible nodes based on
Appendix B MANET Routing Protocols

fractional topology information held in its topology table, using a variation of Dijkstra’s algorithm. Each node reports only part of its source tree $T$ to neighbours so as to reduce overhead. To resolve which section of its source tree to disclose to neighbours, a node must establish its Reportable Node Set - RN. Node $i$ includes node $u$ in RN if it determines that some neighbour $j$ (of node $i$) may select node $i$ to be the next hop (or parent) on the minimum-hop path to destination $u$ [137]. The portion of the source tree $T$ that a node divulges to neighbours is called the Reportable Subtree - RT. RT consists of links $(u,v)$ of $T$ such that $u$ is in RN [137]. To enhance robustness, assuming adequate available bandwidth, a node has the option to reveal its complete source tree $T$. RT is reported to neighbouring nodes in periodic topology updates and topology changes (links up/down) are broadcast more frequently – differential updates. These updates assure rapid dissemination of every topology update to all concerned nodes. Received topology updates are not advanced, but may possibly alter RT, and this change in RT will be reported in the next differential update.

Unlike previous protocols, TBRPF does not use sequence numbers for topology updates. Instead the following technique is used: for each link $(u,v)$ reported by one or more neighbours, only the parent $p(u)$ (a parent node $i$ of node $u$ is a neighbour of $u$ and is the next node on the minimum hop path from node $u$ to the source node) for $u$ is trusted concerning the state of the link. All nodes sustain a topology graph - TG, comprising of trusted links that are recounted by neighbours, and calculates $T$ as the shortest-path tree contained in TG. TBRPF uses RPF to determine minimum hop path routes and minimises overhead using differential HELLO messages. As a consequence of these differential updates much less control traffic is generated and as such this protocol is suited to highly mobile networks with limited bandwidth.

B.1.1.4 Fisheye State Routing – FSR

In FSR [138] the topology update information does not contain information about all nodes. Instead, it periodically broadcasts destination information with a frequency that is dependent upon the destination hop distance (i.e. the scope relative to that destination) thus reducing the update message size. FSR uses the concept of multi-level scope to
Appendix B MANET Routing Protocols

lower routing update overhead in sizeable networks. Update information relating to far destinations is proliferated with a lower frequency than that of near by destinations. Nodes use these updates to determine the network topology and calculate routes. The path that a node uses to send data becomes increasingly more exact as the data advances towards its destination.

FSR uses the fisheye technique (method to reduce the information needed to represent graphical data) proposed by Kleinrock and Stevens [139]. This method is used to define the scope (set of nodes that can be reached within a given number of hops) of each node. In routing terms, the fisheye system is equivalent to preserving exact distance metric and path quality information about the immediate neighbourhood of a node, with increasingly less detail as distance extends. As a consequence of this technique the scalability of the FSR protocol is improved and FSR is suitable for large MANETs with high mobility and limited bandwidth.

B.1.1.5 Landmark Routing Protocol (LANMAR) for Large Scale Ad Hoc Networks

LANMAR builds a hierarchical layer on top of the fisheye scope technique using the concept of subnetting of IP addresses [140,141]. Like subnetting LANMAR splits the whole network into small landmarks, each representing a complete logical subset. These subnets are groups of nodes with a functional empathy with each other and are probably likely to move as a group. Each Landmark is assigned a network id. Each logical subnet has one node acting as the landmark of that subnet. The landmark of each subnet must be aware of all the members in its group. Every node runs a local fisheye algorithm that keeps routes to neighbours up to some hop distance $N$. This distance $N$ is chosen in such a way that if a node is at the centre of a subnet, the scope will cover the majority of the subnet members.

Destination sequence numbers are used by LANMAR to secure loop-free routing. The routes to all landmarks are propagated in the complete network. Every node sustains a distance vector for landmarks of all subnets. Routing of data packets is achieved as follows: if the required destination node lies within the same fisheye scope as the source node the next hop is obtained from the routing table and information is forwarded
directly. If the destination is outside the source scope, then the logical subnet is extracted from the destination and the data is forwarded to the landmark for that particular subnet. Routing table updates are periodic and each node exchanges link state information with neighbours that reside within the same scope. Each node within a subnet has complete knowledge about paths to all other subnet members. Also included in the topology updates are the distance metrics to all other landmarks nodes.

Initially as there are no landmark nodes FSR is used until, with time, a node learns about $N$ neighbours within its scope. This node then declares itself to be the landmark node for its subnet and broadcasts in its topology updates its own Address and Reachable number of subnet members. If more than one node attempts to proclaim itself as a landmark node, the node that can reach the most subnet members becomes the landmark. However, if both nodes can reach the same amount of neighbours then the node with the lowest id is set as the landmark node.

Nodes, due to mobility, can drift outside the fisheye scope of its logical subnet and are referred to as Drifters. The Landmark node of a drifter’s subnet must keep a route to the drifter node. In addition to drifter nodes isolated nodes may also occur. A node is deemed isolated if its logical group is one. If the number of isolated nodes is small (< number of landmark nodes) then these nodes can be treated as individual landmark nodes but if there are a substantial number of these nodes then Lanmar and FSR can be used jointly so as to determine paths [140].

LANMAR extends FSR and exploits group mobility by collecting nodes together in subnets and assigning landmark nodes to each subnet. The LANMAR Protocol provides an efficient and scalable routing solution for MANETs.

**B.1.2 Reactive Routing**

Reactive or on demand routing protocols instigate route discovery when there is a need to establish a communications link between nodes [2, 112]. When source-destination paths have been established route maintenance is used to ensure the continued existence of these routes as routes may break as a consequence of factors such as node mobility or a node powers off due to battery exhaustion. If a link failure is found in a path
then the source node will be notified and may initiate a route discovery procedure to find an alternate path. Bandwidth and battery consumption are less than table driven methods but there is an increase in system delay as the paths between source and destination nodes are found.

B.1.2.1 Ad Hoc On Demand Distance Vector Routing – AODV

AODV routing protocol was designed to build on the DSDV protocol. AODV is an improvement on DSDV as it reduces the number of required broadcasts by creating paths when required i.e. on a demand basis rather than maintaining complete topology tables [142]. AODV is referred to as a pure on demand route acquisition system as nodes that are not on selected paths do not keep routing tables and are exempt from routing table update exchanges. When a node needs to send a packet to some destination node its checks its route table to establish whether it has an existing route to that destination. If so, the information is forwarded to next hop in the direction of the destination. However if does not have a valid path it initiates a path discovery so as to trace the destination. The source node broadcasts a route request (RREQ) packet to its neighbours, which is then forwarded to their neighbours, and so on until it arrives at the destination or some transitional node that has a current path to the required destination. This RREQ packet consists of the source node’s IP address, current sequence number, destination IP address and last known sequence number. Sequence numbers are used to guarantee loop free routing and that routes are determined using the most recent route information. In addition RREQ packets contain a Broadcast ID that is incremented each time a node undertakes route discovery. This ID together with the node’s IP address forms a unique identifier for the RREQ. The source node then broadcasts the RREQ to all its neighbours and waits for a reply. When a neighbour node receives a RREQ it looks at the IP address and broadcast ID to determine if it has seen this RREQ before. For each RREQ received a node keeps a record of the associated IP address and broadcast ID for a particular length of time. If the RREQ has been previously recorded then it is silently dropped otherwise it is processed. Intermediate nodes can reply to a RREQ only if they have a route to the
Appendix B MANET Routing Protocols

destination and the corresponding sequence number is the same or exceeds that contained in the RREQ.

As an RREQ is forwarded intermediate nodes record the addresses of the neighbours that sent the first copy of the broadcast, thus creating reverse route paths. This forwarding process continues until the RREQ reaches the destination node or some transitional node that has a current route to the destination. When a node establishes that it has a current path for the destination, as specified by the RREQ, it creates a RREP (route reply) and unicasts this back towards the neighbour that it received the RREQ from. As the RREP is being reverse routed to the source that initiated the RREQ, nodes along this path record route information pointing to the node that instigated the RREP. As a consequence of this reverse route discovery method AODV can only support bi-directional links. Route table entries are kept for a specific amount of time after which the route entry is erased. As a result of network mobility source nodes must recommence route discovery after each movement to determine a new path to a destination. If an intermediate path node moves then its upstream neighbours becomes aware of this and broadcasts a link failure notification message (an RREP with an infinite metric) [142] to each of its current upstream neighbours that then transmit this link failure to their upstream neighbours until the source node is reached. Again as in previous protocols, HELLO messages are used to establish neighbour connectivity and are broadcast at periodic intervals.

AODV can be likened to an enhanced version of DSDV as it creates routes on demand as opposed to maintaining a list of all possible routes thus minimising protocol overhead. Loop free routing is guaranteed through the use of sequence numbers with only the most recent information being relayed by nodes. Bi-directional links are necessary for the operation of this protocol as route discovery is based on reverse path information.

B.1.2.2 Dynamic Source Routing – DSR

DSR is a source routed on demand protocol with nodes maintaining route caches containing the source routes that they are aware of. Caches are updated if new route or an improved one is found. DSR performs route discovery and route maintenance [143].
Appendix B MANET Routing Protocols

When a node needs to send a data packet it looks up its route cache to see if it already has a destination path. If in the cache it finds an unexpired route it sends the packet using this path. However, if the node does not have such a path it initiates route discovery by broadcasting a RREQ. An RREQ contains the destination address, source address and a unique ID number. Every node that receives this packet checks its cache to see if it has the required destination route. If not, it includes its own address in the packet’s route record and transmits the packet on its outgoing links. Nodes only forward RREQs if the node has not seen the request previously and the node’s address is not contained within the route record. This lessens the number of RREQs that need to be propagated throughout the network for route discovery. A RREP is spawned when either the RREQ arrives at the destination node or at some transitional node that has an unexpired route to the destination. At this stage, contained within the RREQ’s route record will be the sequence of hops from source to destination and this is used to establish the path from source to destination.

For route maintenance DSR uses route error (RERR) packets and acknowledgements (ACKs). When a node encounters a fatal transmission problem it generates a RERR so that a route becomes invalid. Sources receiving this RERR remove the incorrect hop from its route cache and all routes consisting of that hop are curtailed at that point. Sources then select a new route, if available, or else reinitiate route discovery. ACKs are used as a means of verifying accurate operation of route links, and also include a passive acknowledgement in which a node can hear the next hop forwarding a packet along a route.

DSR enables nodes to dynamically discover routes from source to destination with data packet containing the source route, the sequence of hops, in its header. This excludes the need for intermediate nodes to establish routes and these nodes can cache this routing information for themselves. Unlike other protocols DSR does not require periodic broadcasts, for neighbour sensing, routing updates or link status, as this is an exclusively reactive protocol. The absence of periodic updates minimises the protocol overhead and when all network nodes are approximately motionless with respect to each other all communication routes have already been established and node movements that do not adversely affect current routes are not reacted to by the protocol. The operation of
Appendix B MANET Routing Protocols

both route discovery and route maintenance is designed to support unidirectional links as well as bi-directional links.

B.1.2.3 Temporally Ordered Routing Algorithm - TORA

TORA is a distributed, loop free, source-initiated protocol that uncovers multiple roots from a source node to a destination node [144]. TORA is designed to minimise reaction to topological changes. A key feature in its design is that it localises the propagation of control message concerning topological changes to a small set of nodes near the change. To achieve this each node maintains routing information about adjacent nodes (one-hop nodes). This protocol has three major functions: Route creation, Route maintenance and Route erasure.

Routes are established on demand through the use of a query-reply process that builds a directed acyclic graph (DAG) of paths [125, 126] rooted at the destination. As multiple routes are generated, several topological changes necessitate no response, as possessing a single valid route for a destination is adequate. Following any changes that coerce a reaction, the protocol rapidly re-establishes valid routes through a temporally ordered sequence of diffusing computations with each computation consisting of a sequence of directed link reversals. Route creation and maintenance are realised using query (QRY) and update (UDP*) packets. QRY packets are employed by source nodes to seek out nodes with knowledge of an intended destination. This search is implemented by means of flooding. UDP* packets are used in route building and maintenance. Each node has a quintuple associated with it:

- Logical time of a link failure
- The unique ID of the node that defined the new reference level
- A reflection indicator bit
- A propagation ordering parameter
- The unique ID of the node

The first three elements signify the reference level with a new reference level being identified each time a node loses its last downstream link as a result of a link failure. Associated with each node is a height value and these height values are passed in UDP*
Appendix B MANET Routing Protocols

packets during route building and maintenance. A source node broadcasts a QRY packet with the destination node's ID enclosed. A node with a non-null height replies with a UDP* packet containing its height. Upon receiving a UDP* packet a node increases its height value to one more than that of the node that spawned the UDP*. These height metrics are significant for routing as data may only be routed from a node to a node with a lower height. A node with a higher height is considered upstream whereas a lower height is considered downstream. As a consequence of these heights a DAG is constructed from source to destination. Links among neighbours are allotted a direction (directed from higher to lower) as a consequence of neighbour heights.

As nodes move, a DAG is broken and route maintenance is required to regenerate a DAG for the same destination. When the downstream link of a node fails the node generates a new reference level that is propagated by its neighbours with link directions being reversed to mirror the change in adapting to the new reference level as a result of a downstream failure. Route erasure in TORA is performed using clear (CLR) packets. These packets are flooded throughout the network to expunge invalid routes. Route erasure has the effect of setting a node's height to zero.

TORA is a loop free protocol that provides multiple routes to alleviate congestion and reduces algorithmic reactions/communication overhead to conserve available bandwidth and increase adaptability by generating routes only when necessary by creating a DAG rooted at the destination using QRY/UDP packets. Link failures are handled by reversing link directions and the impact of a link failing is reduced as numerous source/destination routes are established.

B.1.3 Hybrid Routing

Hybrid routing protocols [2, 112] combine proactive and reactive protocol techniques. They use distance vector methods in establishing the best path for destination nodes and only disseminate routing information when there is a topological change.
Appendix B MANET Routing Protocols

B.1.3.1 Zone Routing Protocol - ZRP

ZRP is a hybrid protocol as it consists of both proactive and reactive features. This protocol subdivides the network into non-overlapping routing zones for each node and runs independent protocols within and between these zones [145]. ZRP was developed for a particular class of Ad Hoc networks referred to as Reconfigurable Wireless Networks (RWNs). The foremost characteristic of RWNs is an upsurge in node mobility leading to frequent topological changes, increased number of nodes, and a greater network span. ZRP can dynamically modify itself to suit current network conditions by altering its zone radius parameter, which reduces the cost of updates by restricting the scope of these messages to the immediate zone of the change.

Each node zone maintains proactive paths to destinations within their zone (or logical neighbourhood). This zone is branded as a node’s routing zone and is defined as a set of nodes with a minimum distance that is no greater than a parameter that is signified as the zone radius. Nodes whose minimum distance from a source node is exactly equal to the zone radius are termed Peripheral nodes. Within these zones nodes use the Intra-zone Routing Protocol (lARP) [145, 112] to maintain proactive routing information to all zone member nodes. IARP is a proactive protocol that nodes use to discover the topology of its routing zone and as a zone is a subset of the complete network the amount of control traffic necessary is greatly reduced. The main advantage of using IARP is that when there is a change in topology the resulting update messages are propagated within the affected zone only.

For interzone routing ZRP uses the Interzone Routing Protocol (IERP) [145, 112]. This is a reactive protocol that operates between routing zones. If IARP cannot determine the destination then it is assumed that it is beyond the scope of the node’s routing zone and IERP is invoked. The protocol broadcasts a route request to all border nodes within the sprouting zone – this is known as Bordercasting. These nodes then forward the request to nodes outside the routing zone with this process continuing until the destination is found and a reply generated. IERP uses Bordercast Resolution Protocol (BRP) [145, 112] to realise bordercasting.
Appendix B MANET Routing Protocols

ZRP is a hybrid protocol that can be dynamically modified by altering its zone radius so as to adapt to current network operating conditions. Updates concerning topological changes are propagated within the area near change thereby preventing flooding of the entire network. However, topology changes may adversely affect several routing zones.

B.2 Unicast Geographical-Based Routing Protocols

Geographical-based routing protocols [26, 27, 28] depend upon a device being aware of its physical location rather then its relationship with other nodes. As it is both economically and technically possible through the use of GPS for a mobile device to know its location, geographical based routing is feasible.

B.2.1 Geographical Routing Algorithm – GRA

GRA assumes that every node is aware of its geographic position and that of its neighbours [26]. Each node has a routing table in which it stores entries of the form \((p_i, S_j)\), where \(p_i\) is the geographic location of a node and \(S_j\) is the location of one of \(p_i\)'s neighbours. To send a packet between a source node S and a destination node D, node S places in the packet header the geographic location of itself and that of the destination node D. Node S searches its route table to find the nearest neighbour to node D and forwards the packet to that node, where it will again be forwarded towards the destination D. If a node does not possess a route table entry for a necessary destination it initiates a route discovery procedure to find that destination. This route discovery procedure will result in the complete source-destination path being recorded in the route discovery packet header when it arrives at the destination node and this route information is back propagated towards the route discovery initiator node through the use of acknowledgements with intermediate nodes also updating their route tables.
Appendix B MANET Routing Protocols

**B.2.2 Location-Aided Routing – LAR**

LAR is a location based on demand routing protocol that operates in a similar manner to that of DSR [27]. The main difference is that LAR uses location information to limit the broadcast flood area of route discovery packets. These restricted route discovery zones are specified using a rectangular request zone, which contains the source location and the expected destination zone, or the group of nodes that are closer to the destination than the source node. In calculating these zones it is assumed that the source node is aware of the average speed and location of the destination node so that it can determine the expected zone of the destination node. For route discovery the initiator node places position information of itself and the destination in the request packet header that is broadcast within the request zone. Nodes that are within this request zone forward the request with all other nodes discarding this packet. The destination node returns a reply to the request with its current location specified in the packet. If LAR is unsuccessful in discovering a route it resorts to pure flooding throughout the network.

**B.3 Multicast Routing Protocols**

**B.3.1 Multicast Ad Hoc On Demand Distance Vector Routing Protocol – MAODV**

MAODV is an extension to AODV to support multicast routing in ad hoc networks [28]. MAODV also uses sequence numbers to identify freshness of routing information for multicast groups. Each group must elect a multicast group leader that broadcasts periodic GROUP HELLO messages to sustain the group sequence number. For multicast transmissions, nodes form a multicast tree to include all group members. To send a packet to a multicast destination for which a route entry does not exist a node must broadcast a route request throughout the network. Nodes in the multicast tree return a route reply as in unicast routing. Route maintenance is initiated as a response to link breakages with the tree node that is furthest away from the multicast group leader being the node to undertake local route repair.
Appendix B MANET Routing Protocols

**B.3.2 ON Demand Multicast Routing Protocol – ODMRP**

ODMRP is an on demand routing protocol [28, 29] with nodes that wish to transmit to a multicast group establishing group membership by issuing a periodic JOIN REQUEST broadcast that is forwarded by all nodes in the network. When a group node receives such a request it records the forwarding node’s IP address as the next hop for the source node and broadcasts a JOIN TABLE to its neighbour. The neighbour node then scans the table to see if it is the next hop node for the source node in one entry. If it is then the node sets itself as the forwarding node and broadcasts its own JOIN TABLE to its neighbours. The JOIN TABLE is then sent back towards the source node, thereby creating the forwarding group and the route is established for packet transmission.

**B.4 Protocol Performance Evaluation**

The design of efficient routing protocols is a demanding problem in mobile ad hoc networks. Numerous protocols have been proposed in the literature based on varying characteristics and properties, with a number of these have been previously discussed. These protocols have been designed to address the problems of route discovery and maintenance in a MANET environment. Network characteristics, such as network area, node density, node velocity and traffic loads can vary with time so MANET routing protocols must effectively adapt to these changes. MANET protocol performance evaluation is typically based on computer simulation, as the deployment of real world experimental networks is both cost and time prohibitive. Using computer simulation, network scenarios can be easily varied through the use of parameters, such as propagation models, mobility patterns, network size and density, channel access schemes, physical layer modelling, to stress the routing protocol under test.

In order to assess the relative performance of MANET routing protocols [146, 147] have suggested quantitative evaluation metrics that can be used to characterise network behaviour:

- Control Overhead: the number of control traffic packets generated, where control traffic can be route discovery, reply, errors packets and cyclic topology/beaconing
Appendix B MANET Routing Protocols

packets. The amount of control overhead necessary is correlated with the network configuration, e.g. a sparse or highly mobile network may require more control traffic overhead to establish link connectivity whereas a dense or static network may require less.

- Throughput: ratio of the number of data packets generated by source nodes at the application layer and the number of data packets actually delivered to the necessary destination nodes. This parameter signifies the network packet loss and is symptomatic of the packet loss perceived at the application layer.

- Average End-To-End Delay Measurements: measurement of the average time to successfully deliver data traffic, with this measure being recorded at the network layer.

These evaluation metrics are the most prevalent in the research literature however, other metrics that have been used are energy expenditure, bandwidth utilisation and link lifetimes. Several protocol performance evaluations and comparisons have been presented in the research literature, such as [29, 148, 149, 112] but they have used a variety of experimental environments and test conditions, which hinders the direct contrasting of these evaluations.

Both topology-based and geographical-based unicast routing protocols use local neighbour connection information as means of forwarding packets towards the necessary destination. Topology-based protocols employ link state or distance vector updates as a means of establishing a current topological view of the network whereas geographical-based protocols assume GPS capability and use present physical location in generating a geographic reflection of the network topography. The network wide diffusion of routing updates for topology-based routing can be either proactive, reactive or possibly both in the case of hybrid routing whereas for geographical it is always proactive updates that are utilised for ensuring freshness of routing information. Topology-based protocols typically use the shortest path, over which to route traffic, which means selecting nodes that are geographically distant whereas geographical-based protocols opt for the nearest neighbour in terms of physical location. Performance comparisons for topology-based protocols have been presented in [148, 149, 150, 151, 152, 153, 154, 155, 156] with results showing that some protocols outperform others for various topologies, traffic
Appendix B MANET Routing Protocols

flows and mobility rates with DSR and AODV typically exhibiting the most favourable performances.

Hybrid based routing protocols, such as ZRP, use a combination of proactive and reactive techniques in an attempt to optimise protocol performance. For local zone topology maintenance proactive methods are used and interzone routing employs reactive methods and bordercasting to limit redundant forwarding for interzone routing. The performance of ZRP has been investigated in [112] showing that it is a scalable protocol that can be used to achieve an improved performance against protocols that rely solely on proactive or reactive routing.

Multicast routing protocols attempt to reduce the transmission overhead by multicasting to a group rather than using several unicast transmissions in an attempt to reduce channel utilisation, processing delay and packet latencies. MAODV is an extension of the AODV routing protocol that generates a multicast tree on demand to include all group members whereas ODMRP is a mesh-based reactive protocol that uses scoped flooding for packet forwarding. MAODV unlike ODMRP has unicast support. Presented in [29] is a performance evaluation that states that mesh-based multicast protocols out perform tree-based multicast protocols that is substantiated in [28] as it compares the performance of ODMRP with MAODV and states that ODMRP performs better than MAODV.

MANET routing protocol performances vary depending on network topology, traffic loading and mobility patterns but what is obvious from the research work is that no individual protocol is suitable for all environments and each protocol has its own strengths and failings.

B.5 Enabling Technologies

Typically MANET research presupposes that IEEE 802.11b is the underlying enabling technology. However, there are other technologies readily available, such as Bluetooth, ZigBee and ultra wide bandwidth (UWB). IEEE 802.11b offers both contention-free access via infrastructure mode and contention based access using ad hoc mode. MANETs typically rely on contention-based access as this is a point-to-point
Appendix B MANET Routing Protocols

connection whereas infrastructure mode necessitates a fixed-point coordinator (access point). Bluetooth is a low cost industrial specification for wireless personal area networks (PANs) designed primarily for low power consumption, over short distances (10cm – 100m). Bluetooth supports three types of connection: single-slave that enables communication between two devices, multi-slave piconet that allows up to 7 slaves to be connected to one master in an ad hoc fashion and a scatternet, which is the connecting of multiple piconets thus allowing for the establishment of large ad hoc networks. ZigBee was developed to provide a cost-effective, standards-based wireless network that supports low data rates, low power consumption, security, and reliability. The ZigBee Alliance is developing standardised application software on top of the IEEE 802.15.4 wireless standard. ZigBee was designed to be simpler and cheaper than other wireless technologies such as Bluetooth, with the most capable ZigBee node needing in the region of 10% of the software of a Bluetooth node. ZigBee has been developed for low-cost, short transmission range and battery-powered applications such as home and building automation, computer peripherals and medical sensor applications. ZigBee is best suited for low power static networks such as sensor network applications. Ultra wide band is a technology that can transmit data at rates in excess of 100Mbit/s over a wide frequency spectrum with very low power. As a consequence of its low transmit power it transmits negligible interference to existing systems. With UWB technology it will be possible to deliver high-connectivity consumer products in the home, such as video conferencing and home entertainment applications, position location and navigation applications. The IEEE 802.15.3a task group is aimed at developing physical layer standards to support data rates between 110 Mbps and 480 Mbps over short ranges of less than 10 meters. IEEE 802.20 or Mobile Broadband Wireless Access (MBWA) Working Group is looking towards developing a specification for a packet-based air interface designed for IP-based services for low-cost always-on broadband wireless networks. The group will specify physical and medium access control layers of the air interface, operating in bands below 3.5 GHz and with a peak data rate of over 1 Mbit/s.


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


189
Bibliography


Bibliography


Bibliography


