2004

Information dissemination and collection in ad hoc networks

Justin Lipman

University of Wollongong

Recommended Citation
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Information Dissemination and Collection in Ad hoc Networks

A thesis submitted in fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

THE UNIVERSITY OF WOLLONGONG

by

Justin Lipman
Bachelor of Engineering

SCHOOL OF ELECTRICAL, COMPUTER AND TELECOMMUNICATIONS ENGINEERING
2004
Importantly the approaches described in literature only consider the use of a constant (full power) transmission range by all nodes when broadcasting.

We propose two mechanisms: Neighbour Aware Adaptive Power (NAAP) flooding and Localised Minimum Spanning Tree Flooding (LMSTFlood). NAAP utilises a combination of techniques based on neighbour coverage, knowledge of neighbouring relays and the decomposition of high power broadcasts into low power broadcasts through local optimisation to reduce the broadcast transmission range. LMSTFlood utilises the minimum spanning tree (MST) in distributed manner and thus benefits from the properties of the distributed MST and the ability for relay nodes to be self selecting, thus reducing the per packet overhead found in NAAP and other optimised flooding mechanisms. Both mechanisms allow for the transmission range of a broadcast to be adjusted to only include (where possible) those necessary nodes within a broadcast. This limits the effect of broadcasts on neighbouring nodes that may not need to receive a broadcast. The proposed mechanisms are shown to scale significantly better (in terms of reducing the broadcast storm problem and energy consumption) in high node densities than those optimised mechanisms based upon a constant transmission range.

Resource awareness implies that the mechanisms are able to make decisions about which nodes are selected to rebroadcast based upon available resources. These resources may be a node’s available battery power or constraints imposed upon a devices behaviour by a user. The approach of existing optimised flooding mechanisms is to select rebroadcasting nodes irrespective of available resources. We propose two optimised and resource aware flooding mechanisms: Utility based Multipoint Relay (UMPR) Flooding and Utility Based Flooding (UBF). UMPR extends MPR by allowing a limited selection of relays based upon their resources while maintaining the selection of relays with unique neighbours as in MPR. UMPR shows improved performance over existing optimised flooding mechanisms, but through simulation was found to have deficiencies due to the majority of relays selected having unique neighbours. UBF was developed to address these deficiencies. The proposed mechanisms extend the lifetime of the network over successive floods in an energy and user constrained ad hoc network. Importantly, the addition of resource awareness
in the proposed mechanisms does not impact overhead performance compared with existing optimised flooding mechanisms.

Optimised flooding mechanisms reduce problems associated with blind flooding by limiting redundant broadcasts and reducing the effects of broadcasting by reducing transmission power. However, the very nature of Blind flooding and the large number of redundant transmissions ensures higher reliability than optimised flooding mechanisms. We show that the reliability of optimised flooding mechanism are drastically affected by background traffic compared to Blind flooding. This is due to the unreliable nature of IEEE 802.11 broadcasts, which are significantly affected by unicast transmissions and other broadcasts transmissions. We propose Reliable Minimum Spanning Tree (RMST) flooding. RMST is an optimised flooding mechanism that takes advantage of the unique nature of the Minimum Spanning Tree (MST) to replace unreliable broadcast transmissions (as used by existing optimised flooding mechanisms) with more reliable unicast transmissions. Through simulation RMST is shown to have equivalent reliability to Blind flooding with overhead performance exceeding MPR and approaching that of LMSTFlood.

A sensor network is a type of ad hoc network that allows for a node acting as a sink to request information from other nodes. Sensor networks are specialised to optimise the flow of information from nodes back to the sink. In sensor network literature there is much work on mechanisms for information collection. However, much of this work focuses on sensor networks, not on ad hoc networks. We propose Resource Aware Information Collection (RAIC), a distributed resource aware information collection mechanism for ad hoc networks. RAIC utilises optimised and resource aware flooding mechanisms to disseminate requests for information and to create a directed backbone of nodes capable of relaying information back to the sink. RAIC is compared to Directed Diffusion and a brute force approach (Direct Response) and shown to be able to collect information from nodes in the ad hoc network with less overhead in terms of packets transmitted and received, and energy consumption. Additionally given an ad hoc network with resource constrained nodes, RAIC is able to perform significantly more repeated requests for information than Directed Diffusion or Direct Response.
Statement of Originality

This is to certify that the work described in this thesis is entirely my own, except where due reference is made in the text.

No work in this thesis has been submitted for a degree to any other university or institution.

Signed

Justin Lipman
20 January, 2004
Acknowledgments

I wish to thank my supervisors Dr. Paul Boustead and Professor Joe Chicharo for their academic advice and friendship throughout the PhD. I am extremely grateful to Paul for accepting the big responsibility of supervising me, such that he had to spend many hours discussing ideas and reviewing my work. You’ve done a great job! I am also very grateful to Joe for allowing me to do a PhD, providing the necessary scholarship and his overall supervision and guidance towards completion of this thesis. I would also like to thank John Judge for his supervision as my industrial supervisor through the Australian Motorola Research Centre.

Special thanks goes to the ”good” Dr. Mehran Abolhasan and Gaurav Srivastava for many of our technical discussions. I will miss the ”DQ” sessions! I would also like to thank Daniel Franklin for being such a good sport every time I broke my Linux box or beat him at Tetris and for providing quality accommodation and care for the last 6 months - one word, ”Gazpacho”!

Many thanks go to the other members of TITR who made life over these last few years of hard work a great pleasure! The many discussions about problems encountered were also invaluable. I particularly enjoyed my late night tea time chats with Dr. Chun Chung Chou. Special thanks goes to Debbie Farrelly and Farida Nagree for making sure everything always worked.

Finally, I would like to thank my family, Stella Shi and my many friends (The Mcleod Clan, Neil Gomes, Sameer Dixit, Vikram Asthana...) for their support and understanding over the last few years.
Contents

1 Introduction 1

1.1 Background ................................................................. 1
1.2 Thesis Outline ............................................................... 3
1.3 Contributions ................................................................. 7
1.4 Publications ................................................................. 9

2 Literature Review 12

2.1 Introduction ................................................................. 12
2.2 IEEE 802.11 ................................................................. 13
   2.2.1 Medium Access ......................................................... 16
   2.2.2 Hidden Node Problem ................................................. 17
   2.2.3 Unicast and Broadcast Transmission .............................. 19
2.3 Mobile Ad hoc Networks ................................................. 20
2.4 Ad hoc Network Routing Protocols ................................... 22
   2.4.1 Proactive Routing Protocols ........................................ 22
   2.4.2 Reactive Routing Protocols ........................................ 24
2.5 Information Dissemination (Flooding) in Ad hoc Networks ....... 25
   2.5.1 Blind Flooding ....................................................... 26
   2.5.2 Broadcast Storm Problem .......................................... 26
2.6 Optimised and Distributed Flooding in Ad hoc Networks ............. 29
  2.6.1 Heuristic Based flooding ........................................ 30
  2.6.2 Neighbour Coverage Based Flooding .............................. 31
  2.6.3 Cluster Based (CB) Flooding ..................................... 37
  2.6.4 Power Control Based Flooding .................................... 39
  2.6.5 Resource Awareness ............................................... 41
  2.6.6 A Summary of the Features of Distributed Flooding Mechanisms ......................................................... 42

2.7 Reliable Flooding in Ad hoc Networks .............................. 45

2.8 Information Collection in Ad hoc Networks ...................... 49

2.9 Ad hoc Network Simulation ........................................... 55

2.10 Summary .................................................................. 57
  2.10.1 Summary of Open Research Issues Identified in Current Information Dissemination Literature ......................... 58
  2.10.2 Summary of Open Research Issues Identified in Current Information Collection Literature ......................... 59

3 Optimised Information Dissemination .................................. 61

3.1 Introduction ................................................................ 61

3.2 The Cost of Packet Transmission and Reception .................. 63

3.3 Neighbour Aware Adaptive Power Flooding ....................... 64
  3.3.1 NAAP Implementation .............................................. 68
  3.3.2 NAAP Worked Example ............................................ 71
  3.3.3 NAAP Worked Example Results .................................. 76

3.4 Localised Minimum Spanning Tree Flooding ...................... 77
  3.4.1 LMSTFlood Algorithm .............................................. 79
  3.4.2 Calculation of MST .................................................. 81
3.4.3 LMSTFlood Example ............................................. 83
3.5 Simulation Environment ........................................... 84
3.6 Simulation Results .................................................. 85
3.7 Adjusting performance of NAAP ................................... 92
3.8 Conclusions .......................................................... 96

4 Resource Aware Information Dissemination ...................... 99
  4.1 Introduction ....................................................... 99
  4.2 Utility Based Multipoint Relay Flooding ....................... 101
    4.2.1 UMPR - Forwarding Utility ................................ 101
    4.2.2 The UMPR Algorithm ...................................... 103
    4.2.3 Example and Explanation of UMPR flooding ................ 104
  4.3 Utility Based Flooding .......................................... 109
    4.3.1 The UBF Algorithm ........................................ 111
    4.3.2 Example and Explanation of UBF flooding ................ 112
  4.4 Simulation Environment .......................................... 118
  4.5 Simulation Results ............................................... 119
  4.6 Conclusions ....................................................... 123

5 More Reliable Information Dissemination ....................... 127
  5.1 Introduction ....................................................... 127
  5.2 Reliable Minimum Spanning Tree (RMST) Flooding ............. 129
    5.2.1 RMST Algorithm ........................................... 131
    5.2.2 RMST Example ............................................. 133
    5.2.3 RMST Recovery from Failed Unicast Transmissions ............ 133
  5.3 Simulation Environment .......................................... 135
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4 Simulation Results</td>
<td>137</td>
</tr>
<tr>
<td>5.5 Conclusions</td>
<td>143</td>
</tr>
<tr>
<td>6 Resource Aware Information Collection</td>
<td>145</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>145</td>
</tr>
<tr>
<td>6.2 Resource Aware Information Collection</td>
<td>146</td>
</tr>
<tr>
<td>6.2.1 Setup Phase</td>
<td>148</td>
</tr>
<tr>
<td>6.2.2 Capture Phase</td>
<td>150</td>
</tr>
<tr>
<td>6.3 Simulation Environment</td>
<td>151</td>
</tr>
<tr>
<td>6.4 Simulation Results</td>
<td>153</td>
</tr>
<tr>
<td>6.5 Conclusions</td>
<td>158</td>
</tr>
<tr>
<td>7 Conclusions and Future Work</td>
<td>159</td>
</tr>
<tr>
<td>7.1 Overview</td>
<td>159</td>
</tr>
<tr>
<td>7.2 Significant Results</td>
<td>160</td>
</tr>
<tr>
<td>7.3 Further Work</td>
<td>162</td>
</tr>
<tr>
<td>A Verification of Simulations</td>
<td>176</td>
</tr>
<tr>
<td>B Notes on Simulation Randomness, Correlation and Confidence</td>
<td>179</td>
</tr>
<tr>
<td>C Ad hoc Network Topology Generation</td>
<td>182</td>
</tr>
</tbody>
</table>
## List of Figures

2.1 IEEE 802.11 - Infrastructured mode .............................. 15
2.2 IEEE 802.11 - Infrastructureless (Ad hoc) mode .............. 15
2.3 The Hidden Node Problem ........................................ 18
2.4 IEEE 802.11 RTS/CTS and Virtual Carrier Sensing Illustration .... 19
2.5 Ad hoc Network .................................................. 21
2.6 Broadcast Storm Problem - Contention and Redundancy ........ 27
2.7 Optimised Flooding Classifications .............................. 29
2.8 Neighbour Discovery through exchange of beacon messages .... 32
2.9 MPR Flooding and Relay Selection ............................... 34
2.10 Example of a cluster based flood initiated from node A - with only clusterhead and gateway nodes rebroadcasting. .............. 38
2.11 Formation of Relative Neighbourhood Graph using a lune ...... 40
2.12 Relative Neighbourhood Graph .................................. 41
3.1 Neighbour Awareness and Local Optimisation .................... 65
3.2 An example of Blind flooding initiated from a source node (node 1). 71
3.3 An example of Multipoint Relay flooding initiated from a source node (node 1). ................................................. 72
3.4 An example of Neighbour Aware Adaptive Power flooding initiated from a source node (node 1). ................................. 72
3.5 Centralised MST calculated with global knowledge of node topology. 78
3.6 Distributed MST calculated with local one hop topology knowledge. 79
3.7 Hypothetical graph of 4 nodes and their weighted edges 81
3.8 LMSTFlood Example 84
3.9 A comparison of the energy consumed by NAAP, MPR, LMSTFlood, RNG and Blind flooding. 86
3.10 A comparison of packets transmitted by NAAP, MPR, LMSTFlood, RNG and Blind flooding. 88
3.11 A comparison of packets received by NAAP, MPR, LMSTFlood, RNG and Blind flooding. 88
3.12 A comparison of duplicate packets received by NAAP, MPR, LMSTFlood, RNG and Blind flooding. 89
3.13 A comparison of the per packet overhead for NAAP, MPR, LMSTFlood, RNG and Blind flooding. 90
3.14 A comparison of resulting network radius in hops for NAAP, MPR, LMSTFlood, RNG and Blind flooding. 90
3.15 A comparison of the resulting average transmission distance for NAAP, MPR, LMSTFlood, RNG and Blind flooding. 91
3.16 A comparison of the average node coverage per broadcast for NAAP, MPR, LMSTFlood, RNG and Blind flooding. 92
3.17 A comparison of energy consumed by NAAP for varying RangeLimits. 93
3.18 A comparison of packets transmitted by NAAP for varying RangeLimits. 93
3.19 A comparison of duplicate packets received for NAAP with varying RangeLimits. 94
3.20 A comparison of average network radius in hops for NAAP with varying RangeLimits. 94
3.21 A comparison of average transmission distance for NAAP with varying RangeLimits. 95
3.22 A comparison of average node coverage per broadcast for NAAP with varying RangeLimits. 95
4.1 Sigmoid Power Utility $U_p$ 102
4.2 Utility Based Multipoint Relay Flooding Example .................. 105
4.3 Relays with Unique Neighbours ................................. 108
4.4 Utility Based Flooding Example 1 ............................... 113
4.5 Utility Based Flooding Example 2 ............................... 114
4.6 A comparison of flood reachability over multiple successive floods. 120
4.7 A comparison of the number of restricted nodes over multiple successive floods. 122
4.8 A comparison of energy consumed per completed flood vs increasing number of nodes. 123
4.9 A comparison of packets transmitted per completed flood vs increasing number of nodes. 124
4.10 A comparison of duplicate packets received per completed flood vs increasing number of nodes. 124
4.11 A comparison of the additional per packet overhead vs increasing number of nodes. 125
5.1 Distributed MST calculated using local one hop neighbour knowledge 129
5.2 RMST flood utilising IEEE 802.11 unicast and link layer ARQ ........ 134
5.3 RMST Recovery ................................................. 134
5.4 A comparison of flood reachability in the presence of multiple source-destination pairs with increasing CBR traffic. ......................... 137
5.5 A comparison of the energy consumed by nodes for a complete flood with increasing number of nodes. ......................... 140
5.6 A comparison of the packets transmitted by nodes for a complete flood with increasing number of nodes. ......................... 140
5.7 A comparison of the packets received by nodes for a complete flood with increasing number of nodes. ......................... 141
5.8 A comparison of the duplicate packets received by nodes for a complete flood with increasing number of nodes. ......................... 141
5.9 A comparison of the resulting network radius in hops with increasing number of nodes. ......................... 142
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>An example of RAIC Setup and Capture phases</td>
<td>148</td>
</tr>
<tr>
<td>6.2</td>
<td>An example of RAIC Reverse Path Utility</td>
<td>150</td>
</tr>
<tr>
<td>6.3</td>
<td>A comparison of the percentage of nodes information is collected from over multiple successive queries</td>
<td>154</td>
</tr>
<tr>
<td>6.4</td>
<td>A comparison of the number of nodes that enter a minimum energy state over multiple successive queries</td>
<td>154</td>
</tr>
<tr>
<td>6.5</td>
<td>A comparison of the number of nodes that enter a depleted energy state over multiple successive queries</td>
<td>155</td>
</tr>
<tr>
<td>6.6</td>
<td>A comparison of the energy consumed per single query of the ad hoc network.</td>
<td>156</td>
</tr>
<tr>
<td>6.7</td>
<td>A comparison of the packets transmitted by nodes per single query of the ad hoc network.</td>
<td>157</td>
</tr>
<tr>
<td>6.8</td>
<td>A comparison of the packets received by nodes per single query of the ad hoc network.</td>
<td>157</td>
</tr>
<tr>
<td>A.1</td>
<td>Comparison of transmitted packets for a single MPR flood in Glomosim, NS2 and Simplified simulation</td>
<td>178</td>
</tr>
<tr>
<td>A.2</td>
<td>Comparison of received packets for a single MPR flood in Glomosim, NS2 and Simplified simulation</td>
<td>178</td>
</tr>
<tr>
<td>C.1</td>
<td>Average node degree for ad hoc network topology given full transmission range.</td>
<td>183</td>
</tr>
</tbody>
</table>
List of Tables

2.1 A summary of the features of distributed flooding mechanisms. . . . 44

3.1 Simple Comparison of Blind flooding, MPR and NAAP . . . . . . . . 77
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AHBP</td>
<td>Ad hoc Broadcast Protocol</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On Demand Distance Vector</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>BAS</td>
<td>Broadcast Acknowledgement Scheme</td>
</tr>
<tr>
<td>BRG</td>
<td>Broadcast Relay Gateway</td>
</tr>
<tr>
<td>BSET</td>
<td>Broadcast Set</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>CA</td>
<td>Collision Avoidance</td>
</tr>
<tr>
<td>CB</td>
<td>Cluster Based flooding</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CBRP</td>
<td>Cluster Based Routing Protocol</td>
</tr>
<tr>
<td>CD</td>
<td>Collision Detection</td>
</tr>
<tr>
<td>CDS</td>
<td>Connected Dominating Set</td>
</tr>
<tr>
<td>CGSR</td>
<td>Cluster-head Gateway Switch Routing</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSMA</td>
<td>Carrier Sense Multiple Access</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>DBS</td>
<td>Duplicate Broadcast Scheme</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Inter Frame Space</td>
</tr>
<tr>
<td>DS</td>
<td>Distribution Service</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination-Sequence Distance Vector</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
</tbody>
</table>
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>FSR</td>
<td>Fisheye Routing Protocol</td>
</tr>
<tr>
<td>GEAR</td>
<td>Geographical and Energy Aware Routing</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HB</td>
<td>Heuristic Based flooding</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medical</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LAR</td>
<td>Location Aided Routing</td>
</tr>
<tr>
<td>LCT</td>
<td>Local Connectivity Table</td>
</tr>
<tr>
<td>LEACH</td>
<td>Low Energy Adaptive Clustering Hierarchy</td>
</tr>
<tr>
<td>LENWB</td>
<td>Lightweight and Efficient Network-Wide Broadcast</td>
</tr>
<tr>
<td>LMSTFlood</td>
<td>Localised Minimum Spanning Tree Flooding</td>
</tr>
<tr>
<td>LMST</td>
<td>Localised Minimum Spanning Tree</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad hoc Network</td>
</tr>
<tr>
<td>MPR</td>
<td>Multipoint Relay</td>
</tr>
<tr>
<td>MST</td>
<td>Minimum Spanning Tree</td>
</tr>
<tr>
<td>NAAP</td>
<td>Neighbour Aware Adaptive Power</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>NCB</td>
<td>Neighbour Coverage Based flooding</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimised Link State Routing</td>
</tr>
<tr>
<td>PCB</td>
<td>Power Control Based flooding</td>
</tr>
<tr>
<td>RAIC</td>
<td>Resource Aware Information Collection</td>
</tr>
<tr>
<td>RANA</td>
<td>Retransmission After Negative Acknowledgement</td>
</tr>
<tr>
<td>RDMAR</td>
<td>Relative Distance Micro-discovery Ad-hoc Routing</td>
</tr>
<tr>
<td>RMST</td>
<td>Reliable Minimum Spanning Tree</td>
</tr>
<tr>
<td>RNG</td>
<td>Relative Neighbour Graph</td>
</tr>
<tr>
<td>ROAM</td>
<td>Routing On-demand Acyclic Multipath</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request</td>
</tr>
<tr>
<td>RRS</td>
<td>RNG Relay Subset</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>SAR</td>
<td>Sequential Assignment Routing</td>
</tr>
<tr>
<td>SBA</td>
<td>Scaleable Broadcast Algorithm</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>SPIN</td>
<td>Sensor Protocol for Information via Negotiation</td>
</tr>
<tr>
<td>SREP</td>
<td>Sensory Reply</td>
</tr>
<tr>
<td>SREQ</td>
<td>Sensory Request</td>
</tr>
<tr>
<td>STAR</td>
<td>Source-Tree Adaptive Routing</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TopDisc</td>
<td>Topology Discovery</td>
</tr>
<tr>
<td>UBF</td>
<td>Utility Based Flooding</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UMPR</td>
<td>Utility Based Multipoint Relay</td>
</tr>
<tr>
<td>VCS</td>
<td>Virtual Carrier Sense</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WRP</td>
<td>Wireless Routing Protocol</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Since the early 1990s the demand for wireless voice and data communications has increased. This is partially due to the introduction of mobile devices that have grown significantly in computational power and capability. People are attracted by the mobility and flexibility offered. The paralleled growth of the Internet has made available new applications and thus the ability to check one's email, surf the web or share information anywhere and anytime is becoming increasingly desirable. The development of high bandwidth low power communications technologies such as IEEE 802.11 (IEEE, 1997) has removed the barriers found in wired data communication. IEEE 802.11 has made it possible for existing local area networks consisting of fixed infrastructure and fixed points of access to be replaced by wireless local area networks (WLANs) composed of base stations that form cells of coverage and provide a fixed infrastructure without the need for fixed points of access. People are also able to migrate between cells while maintaining connectivity with the network.

However, there exist situations where it may not be possible or feasible to have or to build an infrastructure. A Mobile Ad hoc Network (MANET) (Corson and Macker, 1999) or Ad hoc Network is a new paradigm for wireless communication that allows for nodes to communicate with or without the existence of an infrastructure. The nodes that form an ad hoc network are capable of receiving and transmitting packets
in an ad hoc manner without a base station. More importantly, nodes are able to act as routers thus routing packets between source and destination nodes not within transmission range of each other. Nodes may be constrained by battery power or processing capabilities. Wireless connectivity between nodes may be limited by transmission power, signal attenuation, interference and terrain. Nodes may have varying degrees of mobility, they may switch off or move into or out of range of an ad hoc network thereby changing the ad hoc network topology. Thus ad hoc networks may be characterised by low bandwidth, high error rates, intermittent connectivity and dynamic topology.

The characteristics of ad hoc networks (and the nodes contained within it) create some significant problems. This thesis focuses upon two important areas within ad hoc networking that limit the performance of many applications:

- **Information dissemination** - an one to all process, whereby information flows from one node to all other nodes within the ad hoc network.

- **Information collection** - an all to one process, whereby a node acting as a sink disseminates a request for information throughout the ad hoc network. Nodes receiving the request may then reply to the request with the relevant information.

In ad hoc network literature there has been significant focus upon routing protocols (Johnson, 1994). Routing protocols allow for data to be routed through an ad hoc network between a source and a destination node by utilising intermediate nodes as "stepping stones". Flooding is a type of information dissemination that is vital for the operation of routing protocols. Blind flooding as used by some routing protocols may be used either to find routes between a source and destination node or to disseminate link state information. However, given the broadcast nature of wireless communications and limited bandwidth available, Blind flooding may cause excessive overhead resulting in the Broadcast Storm Problem (Ni et al., 1999). Optimised flooding mechanisms have been proposed to reduce the broadcast storm problem, but suffer as they provide only limited reduction of the broadcast storm problem, are not aware of available resources and are not reliable.
Network management, service discovery and resource discovery in ad hoc networks have not received significant attention. However, as the use of ad hoc networks becomes more prevalent, the need for mechanisms that allow for information to be collected by a single node from all other nodes within an ad hoc network will become more important. Information collection in ad hoc networks relies upon information dissemination to disseminate requests for information and thus suffers from many of the same problems. Additionally, information collection requires efficient mechanisms for the flow of information back to a specific point in the network. While there is significant work in the area of information collection in sensor networks, there is little published work applying these concepts to ad hoc networks.

This thesis focuses upon mechanisms for information dissemination and information collection in ad hoc networks that are:

- optimised to limit the Broadcast Storm Problem thereby improving available bandwidth
- optimised to reduce power consumption given limited battery supplies in mobile devices
- scalable from low to high node densities
- aware of available node resources that may impact performance
- more reliable in the presence of background traffic

1.2 Thesis Outline

In this section, we outline the chapters that comprise this thesis and briefly describe their contents.

In Chapter 2, we introduce the concepts, mechanisms and technologies that comprise ad hoc networking. The problems associated with information dissemination and information collection in IEEE 802.11 based ad hoc networks are described. Existing
literature on mechanisms for information dissemination and information collection that propose to solve these problems are reviewed and categorised.

Chapter 3 introduces Neighbour Aware Adaptive Power (NAAP) flooding and Localised Minimum Spanning Tree Flooding (LMSTFlood). NAAP utilises a combination of techniques based on neighbour coverage, knowledge of neighbouring relays and the decomposition of high power broadcasts into low power broadcasts through local optimisation. LMSTFlood utilises the minimum spanning tree (MST) in distributed manner and thus benefits from the properties of the distributed MST and the ability for relay nodes to be self selecting, thus reducing the per packet overhead found in NAAP and other optimised flooding mechanisms. Both mechanisms utilise transmission power control to reduce their broadcast distance so as to localise the effect of broadcasting and reduce problems associated with the Broadcast Storm Problem. The proposed mechanisms are different from existing optimised flooding mechanisms that only attempt to reduce redundant broadcasts. Existing mechanisms do not account for the power consumption associated with duplicate packet reception and transmitting packets over a distance given that power consumed by a modem’s amplifier is proportional to the square of the distance. In Chapter 3, the performance in terms of the broadcast storm problem and energy consumption of the proposed mechanisms (NAAP and LMSTFlood) are compared to existing optimised flooding mechanisms such as Multipoint Relay Flooding (MPR) (Qayyum et al., 2001) and Relative Neighbourhood Graph (RNG) (Cartigny et al., 2002a). NAAP is shown to have significantly better performance than MPR and approaches that of RNG. Additionally, it is shown that the performance of NAAP may be varied by adjusting the minimum transmission range allowed. LMSTFlood is shown to provide the best performance of the optimised flooding mechanisms. The relative strengths and weaknesses of the proposed mechanisms with respect to their performance and application (such as routing) are discussed. Given the inherent characteristics of NAAP, it may be used by either reactive or proactive routing protocols or as a general information dissemination mechanism. LMSTFlood, however, is seen as not being appropriate for reactive routing protocols, but may be used by proactive routing protocols or as a general information dissemination mechanism.
Existing optimised flooding mechanisms are aimed at limiting the broadcast storm problem. However these mechanisms do not account for the resources available within the ad hoc network and are therefore resource blind in their approach. In Chapter 4, Utility based Multipoint Relay (UMPR) Flooding and Utility Based Flooding (UBF) are introduced. UMPR extends Multipoint Relay flooding by allowing the selection of relay nodes without unique neighbours to be based upon a forwarding utility which is a function of a node’s remaining battery power. This provides only limited selection of relays based upon their resources. UMPR shows improved performance over existing optimised flooding mechanisms, but through simulation was found to have deficiencies due to the majority of relays selected having unique neighbours. UBF was developed to address these deficiencies. In UBF, all relays are selected based solely upon their forwarding utility which may include various utilities that account for a devices remaining battery power or user based constraints. User based constraints may limit a device’s participation in network activities (such as flooding) given low battery power. To determine the performance of UMPR and UBF, both are compared to MPR and Blind flooding. UMPR compared to MPR shows improved performance at extending the lifetime of the network by allowing more successive floods that reach over 90% of the nodes in the network. UMPR when compared to the brute force approach of Blind flooding, shows almost equivalent performance. UBF, which is fully resource aware (all relays are selected based upon their utility), is able to significantly improve upon the performance of UMPR and Blind flooding. Both proposed mechanisms compared to MPR show nearly identical performance in terms of reducing the broadcast storm problem and limiting power consumption.

In Chapter 5, we show that in the presence of an increasing level of background data traffic, the packet delivery performance of existing optimised flooding mechanisms deteriorates to the point of failure. We propose Reliable Minimum Spanning Tree (RMST) flooding. RMST is a reliable optimised flooding mechanism that takes advantage of the unique nature of the distributed Minimum Spanning Tree (MST) to replace unreliable broadcast transmissions (as used by existing optimised flooding mechanisms) with more reliable unicast transmissions. The use of unicast transmissions allows for RMST to be less susceptible to background traffic and provide
comparable packet delivery performance to Blind flooding, yet still achieve optimised flooding performance (in terms of the broadcast storm problem and energy consumption) that surpasses MPR and approaches that of LMSTFlood (discussed in Chapter 3).

In ad hoc networks there may be a need for all-to-one protocols which allow for information collection or "sensing" of the state of the ad hoc network and the nodes that form the ad hoc network. A sensor network (Estrin et al., 1999) is a specialised type of ad hoc network that allows for a node acting as a sink to query other nodes in the sensor network and to collect information in an optimised manner. There is a parallel between this type of sensing in ad hoc networks and that of sensor networks. However, ad hoc networks and sensor networks differ in their application, construction, characteristics and constraints (Akyildiz et al., 2002) (as described in Chapter 2). In Chapter 6, we present Resource Aware Information Collection (RAIC), a distributed information collection mechanism for ad hoc networks. RAIC aims to reduce the overhead associated with information collection in ad hoc networks by taking advantage of neighbour knowledge available to nodes. RAIC makes it possible for a node acting as a sink to collect information from all nodes in an ad hoc network. This process of information collection utilises resource aware flooding (Chapter 4) to create a directed backbone of relays nodes capable of relaying information back to the sink. We show that RAIC compared to Directed Diffusion (Intanagonwiwat et al., 2000) and a brute force approach (Direct Response (Deb et al., 2002)) is able to collect information from nodes in the ad hoc network in a more optimised manner. Additionally, given an ad hoc network with resource constrained nodes, RAIC is able to perform significantly more repeated requests for information than Directed Diffusion or Direct Response.

Finally, Chapter 7 concludes the thesis with a summary of the major results obtained in earlier chapters. A summary of related open research issues in the area of ad hoc networking is presented.
1.3 Contributions

Below is a list of the major contributions of this thesis and the sections in which they appear. Relevant publications are also cited with the contribution.

1. A transmission power control based optimised flooding mechanism is proposed, called Neighbour Aware Adaptive Power (NAAP) flooding (Lipman et al., 2002a) (Lipman et al., 2003d). The use of transmission power control was first proposed in NAAP independently and concurrently with work done in (Cartigny et al., 2002a) on an RNG based flooding mechanism. NAAP makes decisions about rebroadcasting in a distributed manner using two hop neighbour knowledge. NAAP utilises neighbour elimination combined with two novel techniques: neighbour awareness and local optimisation. Neighbour awareness is used to ensure that a relay node only includes its closest neighbour nodes (that it shares with other relays) within its broadcast. The use of transmission power control and local optimisation allows NAAP to decompose a high power broadcast into two or more reduced power broadcasts by selecting a smaller set of closer nodes to include within a broadcast - that will still ensure the propagation of a flood. NAAP is able to adjust its performance by limiting the minimum transmission range used in the local optimisation. NAAP is compared to existing optimised flooding mechanisms and shown to provide significantly better performance than MPR in terms of reducing the broadcast storm problem and energy consumption. (Section 3.3)

2. A distributed optimised flooding mechanism called Localised Minimum Spanning Tree Flooding (LMSTFlood) (Lipman et al., 2003a) that utilises transmission power control is proposed. The MST is used in a distributed manner with local one hop neighbour information to determine those closest neighbours which should be included in any broadcast during a flood. The MST allows nodes upon receiving a packet during a flood to determine whether or not they need to rebroadcast a packet without additional control information, thus resulting in no per packet overhead. LMSTFlood is compared to existing optimised flooding mechanisms (MPR, RNG and NAAP) and shown to provide the
best performance in terms of reducing the broadcast storm problem and energy consumption. (Section 3.4) In addition, in (Lipman et al., 2003a), the performance and ability of selected optimised ad hoc network flooding mechanisms to limit the broadcast storm problem is investigated and the use of transmission power control when broadcasting is identified as having a significant effect on limiting the overhead and improving performance of optimised flooding mechanisms. Furthermore, the use of transmission power control is shown to allow flooding mechanisms to scale as node density is increased. However, the use of transmission power control in flooding introduces additional hops, making its use unsuitable for route discovery mechanisms in routing protocols. (Section 3.6)

3. A resource aware extension to Multipoint Relay flooding is proposed, whereby relays without unique neighbours are selected based upon their forwarding utility which is a function of a relays remaining battery power. The proposed mechanism, Utility based Multipoint Relay (UMPR) Flooding (Lipman et al., 2002b), shows comparable performance in terms of overhead to MPR. Importantly, UMPR is able to extend the number of successive broadcasts in an ad hoc network where nodes have limited battery power compared to MPR and shows comparable performance to Blind flooding - given Blind flooding’s brute force approach. (Section 4.2)

4. A novel optimised flooding mechanism, Utility Based Flooding (UBF) (Lipman et al., 2003c), that selects relay nodes based entirely upon their forwarding utility is proposed. The use of node utilities allows for the selection of relays to be made based solely upon available resources such as battery power and any user based constraints that may define the behaviour of nodes during a flood. In a resource constrained environment, UBF shows significant performance gains over UMPR, MPR and Blind flooding at extending the life of the network thus allowing for significantly more successive broadcasts to be achieved. (Section 4.3)

5. Optimised flooding mechanisms aim to reduce the Broadcast Storm Problem by reducing redundant broadcasts. However, we show in (Lipman et al.,
2004b) and (Lipman et al., 2004c) that improving the optimised performance of flooding mechanisms results in their reliability suffering significantly as background traffic increases. (Section 5.4) We therefore proposed a reliable and optimised flooding mechanism called Reliable Minimum Spanning Tree (RMST) (Lipman et al., 2003a) (Lipman et al., 2004b). RMST is based upon the Minimum Spanning Tree calculated in a distributed manner with local one hop neighbour knowledge. RMST utilises unicast packet transmission as opposed to broadcast packet transmission used by existing optimised flooding mechanisms. RMST is shown to provide comparable reliability to Blind flooding in the presence of background traffic. We show that RMST has equivalent performance at limiting the Broadcast Storm Problem and reducing energy consumption compared to existing optimised flooding mechanisms while having significantly improved reliability. (Section 5.2)

6. An information collection mechanism, Resource Aware Information Collection (RAIC) (Lipman et al., 2003b) (Lipman et al., 2004a), that allows for a single node to collect information from all nodes in an ad hoc network is proposed. RAIC consists of a setup phase and capture phase. RAIC is novel in that it utilises UBF (in the setup phase) to create a backbone (directed acyclic graph) of relays responsible for directing information back to the sink while flooding a request for information. The use of UBF allows for the setup phase to be achieved in a more optimised and resource aware manner than existing mechanisms developed for sensor networks. Furthermore as information travels back towards the sink, RAIC utilises a reverse path utility calculated during the setup phase that summarises the benefit of various possible paths back to the sink. Thus packets travel via the best path available back to the sink. RAIC, in a resource constrained environment, is shown to have significantly better performance than existing mechanisms. (Section 6.2).

1.4 Publications

Publications arising from work directly related to this thesis are listed below:
1. J. Lipman, P. Boustead, J. Judge, "Efficient Scalable Information Dissemina
tion in Mobile Ad-hoc Networks". In proceedings of AD-HOC Networks and Wireless, Fields Institute (Toronto), September 2002, pages 119-134.


Publications submitted for review:

Chapter 2

Literature Review

2.1 Introduction

In the last two decades wireless data communications has been an active area of research. It began with the US military exploring applications of packet radio networks during the eighties. This research led to the development and adoption in the nineties of new wireless technologies capable of providing high bandwidth mobile data communications. The combination and popularity of new technologies such as mobile voice communication, mobile computing devices with increased processing capabilities and battery life, and the Internet has lead to increased demand from consumers who want to be "mobile" and "connected". Additionally, the necessity for devices to communicate in a wireless manner has resulted in the development of wireless data technologies such as IEEE 802.11 (IEEE, 1997), Bluetooth (Bluetooth, 2002) and HiperLAN (ETSI, 1998). This has led to the development of new paradigms for networking such as Mobile Ad hoc Networks (MANETs) (Corson and Macker, 1999), alternatively called "Ad hoc Networks".

In this chapter we review the technology and issues associated with ad hoc network communication that range from the medium access layer to the application layer. The IEEE 802.11 wireless standard for communication is explored and the issues associated with wireless communication such as the "Hidden Node Problem" are described. The differences between broadcast and unicast communication in IEEE 802.11 are
discussed. We explore "Ad hoc Networking" and look at many of the issues associated with communication in such environments. We then look at applications of ad hoc networks and the routing protocol mechanisms that allow applications to operate in an ad hoc network environment. The problems associated with dissemination of information throughout ad hoc networks, specifically the "Broadcast Storm Problem" are explored and distributed mechanisms that limit the broadcast storm problem through optimisation are reviewed. We discuss the importance of resource awareness and describe existing mechanisms that have an element of resource awareness. Improving the reliability of information dissemination is discussed and more reliable mechanisms relevant to IEEE 802.11 ad hoc networks are reviewed. Finally, we consider information collection in a specialised type of ad hoc network called a "Sensor Network". We review existing literature in sensor networks that is relevant to information collection in IEEE 802.11 ad hoc networks.

2.2 IEEE 802.11

A wireless local area network (WLAN) is a network where the nodes within a network communicate over wireless links. The nodes are not bound to physical connections as with a local area network (LAN). The IEEE 802.11 standard for WLANs was first adopted in 1997 (IEEE, 1997). The standard defines a medium access control (MAC) sublayer, MAC management protocols and services, and three physical layers at 1 Mbps and 2 Mbps. The three defined layers are line of sight Infra-Red (IR) baseband and two radio layers at 2.4 Ghz - Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). Three new physical layers, IEEE 802.11a (IEEE, 1999b), IEEE 802.11b (IEEE, 1999a) and IEEE 802.11g (IEEE, 2003) have recently been adopted by the IEEE. IEEE 802.11b extends IEEE 802.11 DSSS allowing for 11 Mbps data rates within the 2.4Ghz ISM band. IEEE 802.11a is based upon Orthogonal Frequency Domain Multiplexing (OFDM) and achieves 54 Mbps data rate within the 5Ghz UNII band. IEEE 802.11g like IEEE 802.11a is based upon OFDM, however it operates within the 2.4Ghz ISM band and is backward compatible with IEEE 802.11b.
There are a number of characteristics that are unique to a WLAN (when compared to a LAN) that the IEEE 802.11 standards take into consideration. The physical characteristics of a WLAN introduce range limitations and unreliable media, dynamic topologies where nodes roam between base stations, interference from outside sources, and lack of the ability for every device to hear every other device within the WLAN. These limitations forced the WLAN standards committee to create fundamental definitions for short-range LANs made up of components that are within close proximity of each other. Larger geographic coverage is handled by building larger LANs from the smaller fundamental building blocks or by integrating the smaller WLANs with an existing wired network.

In the IEEE 802.11 specification there are two modes of operation: Infrastructured and Infrastructureless (Ad hoc). Infrastructured mode (Figure 2.1) is based upon a cellular architecture where a network may be composed of one or more cells, each called a Basic Service Set (BSS). An Access Point (AP) or Base Station is responsible for controlling each BSS. APs control the flow of packets. A typical WLAN may consist of one or more BSS. In the case where more than one BSS exists, the APs are connected to each other via a backbone called the Distribution System (DS). The DS is typically a wired backbone, however it is possible for this to be a wireless backbone. The DS is responsible for allowing all BSS and AP to appear as a single WLAN, thus a node may migrate between BSS without issue. In Figure 2.1, AP1 and AP2 are access points connected via a DS. Nodes N1, N2, N3, N4, N5 communicate with each other through AP1 and AP2. Node N3 is shown migrating from AP1’s BSS to AP2’s BSS.

In ad hoc mode (Figure 2.2), a set of nodes within transmission range of each other are able to form a network and communicate directly with one another. Nodes not within transmission range are unable to communicate with each other. Thus in Figure 2.2 node N1 is able to communicate with both nodes N2 and N3, however nodes N2 and N3 are not within transmission range and are therefore unable to communicate with each other.
Figure 2.1 IEEE 802.11 - Infrastructured mode

Figure 2.2 IEEE 802.11 - Infrastructureless (Ad hoc) mode
2.2.1 Medium Access

In IEEE 802.11 the basic medium access mechanism implemented at the MAC layer is Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) (IEEE, 1997). A node utilising CSMA that has a data frame to transmit will first sense the shared medium. This sensing is performed by listening for any existing transmissions. If the medium is busy, then the station will delay its transmission. However, if the medium is not busy, then the node will begin transmitting the data frame. CSMA mechanisms are useful in scenarios where the shared medium is lightly loaded (low traffic) as there is minimal delay prior to transmission. The problem with CSMA is that if the shared medium is heavily loaded (high traffic), then the probability of nodes simultaneously sensing the medium as being free and then transmitting increases. Thus the possibility of collisions occurring increases.

In wired networks, such as IEEE 802.3, Carrier Sense Multiple Access / Collision Detection (CSMA/CD) is implemented. CD requires that the transceiver of the transmitting node be full duplex, thus able to both transmit and receive simultaneously. If a collision is detected by a transmitting node, then it delays before rebroadcasting using a random exponential back-off. In wireless communication, CD mechanisms are restrictive for two reasons:

- CD assumes that all nodes are able to detect each other’s transmissions, however in a wireless environment this is not always the case. A node may sense the medium as being free because it is unable to detect another transmission and may possibly then transmit thus resulting in a collision at a neighbouring node that is already receiving a transmission.

- Implementing a CD mechanism requires full duplex transceivers, which consume more energy due to additional electronics and increase the price of the wireless transceiver significantly.

Given the CD mechanisms are not viable in a wireless environment, IEEE 802.11 combines a Collision Avoidance (CA) and positive acknowledgements (based on Stop and Wait). Utilising these two mechanism the following scenario occurs:
• **Transmitting Node:** If a node is about to transmit, the MAC will first sense the medium. If the medium is busy, then the transmission is delayed. If the MAC senses the medium is free for a specified period (Distributed Inter Frame Space - DIFS), then the transmission will occur. The transmitting node will wait for a positive acknowledgement (ACK) from the receiving node, thus verifying that the transmission was received correctly. If a no ACK is received, the transmitting node will retransmit the frame a specific number of times before discarding the frame.

• **Receiving Node:** A node upon successfully receiving a transmission will verify the CRC of the received frame and send an acknowledgement (ACK) to notify the sender. If the frame is corrupted, then no ACK is transmitted.

The combination of CA with a positive acknowledgement in IEEE 802.11 medium access provides a means of reducing the probability collisions by nodes wishing to transmit data. However, the mechanism requires that nodes be able to overhear other broadcasts. In a situation where two nodes are unable to hear each other, the mechanism is not as effective. This is referred to as the *Hidden Node Problem.*

### 2.2.2 Hidden Node Problem

Figure 2.3 shows the *Hidden Node Problem* experienced with IEEE 802.11 in ad hoc mode. This is the result of two nodes A and C that are not within direct transmission range of each other transmitting a message to a shared neighbouring node (node B). In IEEE 802.11 both node A and C prior to broadcasting will sense the medium to determine if it is free. If both nodes A and C transmit at the same time or if one node transmits while the other is still transmitting, then a collision will occur at the receiving node B.

IEEE 802.11 avoids the hidden node problem by utilising a Virtual Carrier Sense (VCS) mechanism as shown in Figure 2.4. The VCS is used to reduce the probability of a collision caused by neighbouring nodes (within transmission range) of the receiver that may possibly wish to transmit a frame. In VCS, a source node’s MAC initially transmits a Request To Send (RTS) frame prior to transmitting a data frame.
The RTS specifies the destination that the data is destined for (and an approximate period required for the transmission). Upon receiving the RTS, the destination node’s MAC transmits a Clear To Send (CTS) frame authorising the source node to transmit the data frame. All neighbouring nodes receiving the RTS and/or CTS set their Virtual Carrier Sense indicator called Network Allocation Vector (NAV) for the given duration. This information is then used by all neighbouring nodes when performing future carrier sensing. The source node is then able to transmit the frame without a packet collision caused by neighbouring nodes of the destination node.

In Figure 2.4, the Short Inter Frame Space (SIFS), is used to separate transmissions belonging to a single frame and is the minimum inter frame space. The value is fixed at 28 microseconds and is calculated to account for the transmitting node’s transceiver switching from transmit mode back to receive mode in order to receive and decode an incoming packet. The Distributed Inter Frame Space (DIFS), is the inter frame space used when a station begins a new transmission.

The exchange of RTS/CTS for VCS is dependent on the size of the data frame. As the RTS (20 bytes) and CTS (14 bytes) frames are small there is a lower probability
of a collision occurring compared to a data frame (if the data frame is much larger than the RTS/CTS frame). However, if a data frame is small in size, then the standard allows for short packets to be transmitted without RTS/CTS.

### 2.2.3 Unicast and Broadcast Transmission

IEEE 802.11 provides two mechanisms for data transmission to one or more neighbouring nodes: unicast and broadcast. It is important to note that significant differences exist between unicast and broadcast data transmission in IEEE 802.11.

Unicast (one to one) transmission allows a source node to send data directly to one destination node within transmission range. The MAC layer utilises CSMA/CA with RTS/CTS and positive acknowledgements. If a frame is not received or the cyclic redundancy check (CRC) fails verification, then a retransmission will occur as no positive acknowledgement is received. The use of CSMA/CA, RTS/CTS and positive acknowledgements allows unicast transmission to be less susceptible to collisions, packet loss and the hidden node problem.

Broadcast (one to all) transmission allows a source node to send data directly to all nodes within transmission range. The medium access mechanism, CSMA/CA described earlier is not used in its entirety. Prior to data transmission, carrier sensing (CS) is performed by the MAC. If the medium is free then the MAC will transmit the data frame. If the medium is not free, then random back-off delay occurs. The MAC
layer does not exchange RTS/CTS frames with surrounding nodes prior to transmission. The received data frame's CRC is verified upon reception at each receiver and allowed to progress up the protocol stack. If the data frame fails CRC verification, it is deleted. Unlike unicast transmission, no positive acknowledgement is transmitted back to the source. The broadcasting node has no mechanism to determine if a broadcast was received by one or all nodes. Given the lack of acknowledgements in broadcast transmission, no data frame retransmissions occur. Thus unicast transmission is more reliable than broadcast transmission.

In IEEE 802.11, there is a difference between broadcast packets and unicast packets at the IP layer and MAC layer. Broadcast packets are addressed to an IP address of 255.255.255.255 (MAC Address = FF:FF:FF:FF:FF:FF) and hence are received by any node within broadcast range. Unicast packets are directed towards a single node and therefore use a unicast address and the corresponding MAC address of the destination node. The IEEE 802.11 MAC layer will not allow a packet to ascend the TCP/IP stack unless the destination MAC address is its own or a broadcast address.

2.3 Mobile Ad hoc Networks

The adoption of high bandwidth low power communications technologies such as IEEE 802.11 (IEEE, 1997) has made it possible for existing local area networks consisting of fixed infrastructure and fixed points of access to be replaced by WLANs composed of base stations that form cells of coverage and provide a fixed infrastructure without the need for fixed points of access. However, there exist situations where it may not be possible or feasible to have or to build an infrastructure. A Mobile Ad hoc Network (MANET) (Corson and Macker, 1999) or Ad hoc Network as shown in Figure 2.5 is a new paradigm for wireless communication that allows for nodes to communicate with or without existing infrastructure. An ad hoc network may operate in isolation or be connected to a fixed network (such as the Internet) via a base station (gateway). Ad hoc networks lack the centralised administration or standard support services regularly available on conventional networks.

Figure 2.5 consists of an ad hoc network attached to the Internet via a base station
The base station is not necessarily an IEEE 802.11 access point, but merely a node acting as a gateway with access to the Internet. If node N6 wishes to communicate with node N4 in the Ad hoc network, it must rely upon the use of a routing protocol to communicate over multiple hops (N6, N5, N3, N4).

The nodes that form an ad hoc network are capable of receiving and transmitting packets in an ad hoc manner without a base station thus utilising the ad hoc mode of IEEE 802.11. More importantly, nodes are able to act as both an edge device and a router, thus they are able to route packets between source and destination nodes not within transmission range of each other. Nodes may be constrained by battery power or processing capabilities. Nodes may have varying degrees of mobility, they may switch off or move into or out of range of an ad hoc network thereby changing the ad hoc network topology. Wireless connectivity between nodes may be limited by transmission power, signal attenuation, interference and terrain. Thus ad hoc networks may be characterised by low bandwidth, high error rates, intermittent connectivity and dynamic topology.

Ad hoc networks may be employed in many scenarios and are particular useful in dynamic network environments where the topology of the network may change con-
Literature Review

...continuously. They are also useful in areas where a networking infrastructure may not be easily implemented. Some typical applications of ad hoc networks are:

- Home, office and personal area networks
- Disaster Recovery
- Conferences
- Heritage buildings
- Military battle field operations

2.4 Ad hoc Network Routing Protocols

Ad hoc network routing protocols form the basis of communication in ad hoc networks. Routing protocols are responsible for delivering packets between nodes not within broadcast range. This requires the use of cooperative intermediate nodes that are able to act as routers in a distributed manner, thus allowing for data packets to be forwarded towards their destination. Ad hoc network routing protocols may be classified based upon how they determine routes into three groups: proactive, reactive and hybrid. Hybrid routing protocols utilise a combination of proactive and reactive routing. Routing protocols may further be classified depending on whether they maintain a flat network structure or a hierarchal network structure. In this section, routing protocols are briefly described with an emphasis on how they disseminate control information.

2.4.1 Proactive Routing Protocols

Proactive routing protocols (Broch et al., 1998) require that each node maintain route information to every other node in the network. Route tables are periodically or dynamically updated if the network topology changes. Proactive routing protocols differ in how they detect changes in network topology, how they maintain route tables and how they disseminate this information to other nodes in the network. Proactive routing protocols experience minimal delay when routing packets as routes are
available immediately from constantly maintained route tables. Although there is no initial penalty when a route to a destination is required, there is a constant overhead associated with disseminating link state or route table information throughout the network. This results in a reduction in network capacity due to constant and possibly heavy control traffic delivery. This is made worse in the presence of node mobility. Additionally, proactive protocols do not scale effectively as node density and node numbers increase (Santivez et al., 2001). Constant dissemination of control information throughout the ad hoc network also results in increased power consumption.

The majority of proactive routing protocols disseminate control information throughout the ad hoc network use blind flooding (as described later in this chapter). Blind flooding is shown to result in the broadcast storm problem (Ni et al., 1999) and is thus not efficient. Examples of proactive routing protocols that utilise blind flooding are Destination-Sequenced Distance Vector (DSDV) (Perkins and Bhagwat, 1994) and Wireless Routing Protocol (WRP) (Murthy and Garcia-Luna-Aceves, 1995). Other proactive routing protocols such as Fisheye State Routing (FSR) (Gerla, 2002) limit the rate at which they update route information depending on the distance. Routes to closer nodes are maintained more regularly, whereas routes to remote nodes are maintained less regularly. Source-Tree Adaptive Routing (STAR) (Garcia-Luna-Aceves and Spohn, 1999) eliminates periodic dissemination of control information in favour of conditional dissemination, thus reducing the constant overhead. However, blind flooding is still required. In Cluster-head Gateway Switch Routing (CGSR) (Chiang et al., 1997) a hierarchy is created based upon node clustering. Cluster-heads control the flow of route information within their cluster and between clusters, thus reducing the amount of route information and limiting the dissemination of the route information. The Optimised Link State Routing (OLSR) (Jacquet et al., 2000) protocol attempts to reduce the problems associated with blind flooding by utilising an optimised flooding algorithm called Multipoint Relay (MPR) (Qayyum et al., 2001) flooding. The use of an optimised flooding algorithm reduces the problems associated with blind flooding and allows OLSR to scale more effectively given an increased number of nodes.
2.4.2 Reactive Routing Protocols

Reactive routing protocols (Broch et al., 1998) are designed to reduce the overheads associated with proactive routing protocols. They do this by only maintaining information for active routes. Reactive routing protocols do not proactively maintain routes to all nodes, therefore they must perform route discovery when a route to a destination node is required. Route discovery requires that a "route request" (RREQ) packet be blind flooded throughout the network. When the destination (or a node with an active route to the intended destination) receives the RREQ a "route reply" (RREP) is sent back to the source of the route request. The RREP may either be blind flooded back to the source or it may be unicast back along the path followed by the RREQ. As routes are not immediately available, reactive protocols have a much higher initial delay at the start of communication than proactive routing protocols. Given that flooding forms the basis of route discovery, reactive routing protocols are effected by the broadcast storm problem. This is made worse by: increasing node density, higher node mobility and the number nodes of performing route requests for peer to peer communication. Importantly for mobile devices, there is no constant power usage due to periodic flooding of link state or route table information as with proactive routing protocols.

Both Dynamic Source Routing (DSR) (Johnson and Maltz, 1996) and Ad hoc On-Demand Distance Vector Routing (AODV) (Perkins and Royer, 1999a) protocols utilise blind flooding as a means of performing route discovery. However, they differ in the way they maintain routes to destination nodes and also in the amount of information required to route packets. To reduce the effects of blind flooding, these protocols use route caching as well as limiting the number of hops for route discovery. The Routing On-demand Acyclic Multi-path (ROAM) (Raju and Garcia-Luna-Aceves, 1999) protocol limits the effects of flooding by using directed acyclic subgraphs based upon distance between the source and destination for the propagation of a flood. This eliminates the propagation of a flood in a direction along a subgraph if the destination is not reachable along that subgraph. In Relative Distance Micro-discovery Ad-hoc Routing (RDMAR) (Aggelou and Tafazolli, 1999), overhead associated with route discovery is reduced and localised by limiting each RREQ packet
to a certain number of hops. However, this localisation of route requests can only occur if the source and destination node have communicated before and exchanged position information. If the nodes have not communicated before, then the route request is not localised. Location Aided Routing (LAR) (Ko and Vaidya, 1998) requires that nodes have a GPS device and therefore are aware of their location. Thus overhead associated with route discovery is reduced by limiting the direction and scope of flooding. This protocol defines zones specifying which direction a RREQ packet may travel towards. RREQ packets therefore only travel in the approximate direction of the intended destination. Cluster-Based Routing Protocol (CBRP) (Jiang et al., 1999) is a hierarchal routing protocol based upon clustering. Clusterheads are defined and responsible for the nodes within each cluster. To reduce the effects of route discovery, only clusterheads exchange and propagate RREQ packets.

2.5 Information Dissemination (Flooding) in Ad hoc Networks

In ad hoc networks the process of information dissemination from one node to all nodes is commonly referred to as "flooding" or "broadcasting". As the usage of these terms is not consistent within the ad hoc research community we provide the following definitions:

- **Broadcasting**: The process of sending a packet of information to all neighbouring nodes that are directly reachable within 1-hop. These nodes are not expected to rebroadcast or relay the packet to others.

- **Flooding**: The process of disseminating a packet of information to all nodes within an ad hoc network. As not all nodes will be reachable directly, 1-hop neighbouring nodes are expected to relay the packet of information to other nodes using a flooding algorithm.

Flooding forms the basis of nearly all communications in ad hoc networks. As described previously, routing protocols allow peer to peer communications between
nodes in an ad hoc network. In order to set up this peer to peer connection between two nodes, a routing protocol may either proactively determine the best path or discover the path reactively. Proactive routing protocols can benefit greatly from optimising the process of flooding as this can significantly reduce overhead associated with disseminating link state or route table information. In the reactive approach, nodes do not need to disseminate link state or route table information. However, reactively discovering paths may lead to intervals of very high network activity due to flooding when multiple nodes perform route discovery. As with proactive routing, reactive routing may also benefit from optimising the process of flooding.

2.5.1 Blind Flooding

The simplest mechanism for information dissemination within a network is Blind flooding. Blind flooding is used by routing protocols such as AODV (Perkins and Royer, 1999b) and DSR (Johnson et al., 2001) to perform route discovery. Blind flooding may also be used in network management to distribute state information or in zero start auto-configuration. In blind flooding, a node broadcasts a packet, which is received by its surrounding neighbours. Each receiving neighbour then verifies that it has not broadcast the packet before. If not, then the packet is rebroadcast. Blind flooding terminates when all nodes have received and rebroadcast the packet. In blind flooding there is a need for broadcasting nodes to introduce a random amount of delay (jitter) prior to broadcasting. This is to aid in removing collisions from co-located nodes which may rebroadcast at the the same time after receiving the same message. Blind flooding always chooses the shortest path, because it chooses every possible path in parallel. Therefore no other algorithm can produce a shorter delay. Of course this is not quite accurate, as in wireless networks blind flooding results in the broadcast storm problem which may delay or inhibit the propagation of a blind flood.

2.5.2 Broadcast Storm Problem

The broadcast storm problem (Ni et al., 1999) may increase resource contention and hence impede the floods overall performance. The broadcast storm problem states
that, in a CSMA/CA network, flooding is extremely costly and may result in the following:

- **Redundant rebroadcasts** - occurs when a node decides to rebroadcast a message to its neighbouring nodes and all its neighbouring nodes have already received the message.

- **Medium Contention** - occurs when neighbouring nodes upon receiving a message decide to rebroadcast the message. They may need to contend with each other for the broadcast medium.

- **Packet Collision** - because of the deficiency of the back-off mechanism, the lack of RTS/CTS dialogue, and the absence of CD, collisions are more likely to occur and cause more damage (lost or corrupted messages).

In Figure 2.6 the problems of redundant broadcasting and contention as stated by the broadcast storm problem are shown when performing a blind flood. In Figure 2.6, if node A initiates a flood of a message and the message is received by nodes B and
C. These nodes according to blind flooding are required to rebroadcast the message if they had not rebroadcast it before. Therefore nodes B and C will rebroadcast the message. Node D will receive the message and also rebroadcast the message. The following will result:

1. Although node A initiated the flood by broadcasting the message, it is still within broadcast range of nodes B and C. Therefore it will receive two redundant copies of the message from nodes B and C. This is also the case with nodes B and C each receiving the broadcast from node D and from each other.

2. Nodes B and C must contend for the broadcast medium as shown in Figure 2.6 by the shaded area. If there are more nodes within the area, there will be an increase in contention for the broadcast medium.

3. Node D will receive a total of two broadcast messages (one each from nodes B and C).

From Figure 2.6 it can be seen that only two broadcasts (nodes A and B or nodes A and C) are actually necessary for all nodes to receive the broadcast from node A.

An additional problem with broadcasting in IEEE 802.11 is that the lowest common denominator bit-rate is employed for broadcasting. In IEEE 802.11 this is 1Mbps as opposed to the highest bit-rate being used for unicast transmission. The result of this is that broadcasts affect a larger radius than unicast transmissions and may therefore corrupt unicast data packets. This highlights the reasons for reducing redundant transmissions. It also makes an argument for using reduced transmission power for broadcasting to reduce the transmission distance.

Flooding forms an integral part of all communication in ad hoc networks as it forms the basis of route discovery in reactive routing protocols and as a means of disseminating link state or route table information in proactive routing protocols. It may also form the basis of many other operations in ad hoc networks from autoconfiguration to multicasting. Flooding may be seen as a bottleneck in limiting the capacity of ad hoc networks to support services.
2.6 Optimised and Distributed Flooding in Ad hoc Networks

Given the problems with the Broadcast Storm Problem, it is important in ad hoc networks to utilise mechanisms to limit its effect. Optimised flooding algorithms provide a means of limiting the broadcast storm problem as they reduce redundant broadcasts, medium contention and packet collisions. Additionally they may also reduce power consumption.

In Figure 2.7 we classify optimised flooding mechanisms as either centralised or distributed. Centralised mechanisms require global topology information and attempt to determine energy efficient optimal broadcast trees. The creation of these broadcast trees is NP-Hard where omni-directional broadcasts are used (Li and Nikolaidis, 2001). To solve the NP-hard problem, mechanisms based upon heuristics have been proposed. In (Wisielthier et al., 2000) three centralised mechanisms are proposed:

- **Broadcast Incremental Power** - Forms a broadcast tree using a modified
version of Prim's algorithm (Prim, 1957) that accounts for and takes advantage of the broadcast nature of communication in ad hoc networks.

- **Broadcast Least Unicast cost** - Forms a minimum energy tree by the superposition of the least cost unicast paths to each destination node in the ad hoc network.

- **Broadcast Link based Minimum Spanning Tree** - Forms a centralised minimum spanning tree where the link cost is not distance but the required transmission energy to reach the destination node.

Given the dynamic nature of ad hoc networks, a centralised approach is not desirable nor feasible as the cost of obtaining global topology information is restrictive in terms of overhead. Thus in this thesis we focus upon mechanisms that utilise a distributed approach whereby decisions of whether or not to rebroadcast are made based upon localised information at each node. As shown in Figure 2.7, we propose to classify distributed flooding mechanisms into **Simple Heuristic Based** flooding, **Neighbour Coverage Based** flooding, **Cluster Based** flooding and **Power Control Based** flooding.

### 2.6.1 Heuristic Based flooding

Heuristic Based (HB) flooding mechanisms attempt to limit the broadcast storm problem by making rebroadcast decisions based upon a heuristics. Heuristic based flooding methods require careful selection of parameters and thresholds and are therefore closely related to the ad hoc network environment. Their performance as such is dependent on the resulting parameters and thresholds as utilised in the heuristic.

In (Ni et al., 1999) and (Tseng et al., 2001) several heuristic based flooding mechanisms are proposed:

- **Counter based** - the decision to rebroadcast is based upon a threshold value for the number of duplicate packets received by the broadcasting node. If the number of duplicate packets is less than the threshold value then the node
will rebroadcast. Otherwise it will not rebroadcast. An expected additional coverage function may be defined, which shows that the more times a host has heard the same broadcast packet, the less additional coverage the host provides if it rebroadcasts the packet.

- **Distance/Location based** - the heuristic may involve distance in a relative sense - physical distance between nodes or the transmission power required. Each node is equipped with a GPS device or is able to determine a neighbouring nodes signal strength. Given the distance or location of broadcasting nodes it is possible to calculate the expected additional coverage (in terms of area) a node may provide by rebroadcasting.

- **Probability based** - the decision to rebroadcast is based upon a random probability heuristic. This probability may be as simple as flipping a coin or it may be more complex involving probabilities based upon other parameters that may be useful in determining whether or not to rebroadcast such as node density, duplicate packets received, battery power or a nodes participation/benevolence within the network.

### 2.6.2 Neighbour Coverage Based Flooding

In Neighbour Coverage Based (NCB) flooding, nodes periodically or dynamically broadcast beacon messages to advertise their own existence and also discover the existence of neighbouring nodes within transmission range (one hop). Beacon messages may typically contain the broadcasting node's address and any neighbouring nodes that the node may be aware of. Thus allowing for neighbour topology up to two hops to be determined. The use of neighbour information allows the link state topology of nodes to be determined. It is also useful in situations where GPS may not work such as indoors. The exchange of beacon messages allows for additional information about neighbouring nodes to be exchanged. This additional information may be a node's remaining battery power, any user-based constraints, physical coordinates acquired through a GPS device, signal to noise ratio (SNR) measurements (acquired from the MAC layer) and possible device characteristics such as maximum broadcast power.
However, the use of beacons for neighbour discovery may suffer from various problems: (i) Consider Figure 2.8, where the average neighbour degree for each node is \( n \). A node wishing to discover its local two hop topology must first wait for each of its \( n \) one hop neighbours to receive beacons from their \( n \) one hop neighbours. Thus to discover two hop topology requires the exchange of at least \( n^2 \) beacons. (ii) Nodes do not transmit beacons simultaneously, thus multiple exchanges of beacons over an extended period of time may be necessary to discover two hop topology. (iii) As beacons are sent using broadcast packets, there is a possibility of packet collisions resulting in loss of the beacon. This is shown in Figure 2.8 where beacon messages from nodes D and E collide at node C. Node C therefore only shows the existence of node F when it sends a beacon message to node A. (iv) Varying degrees of node mobility may result in link state errors. (v) Exchange of link state information will increase the packet size of the beacon making it more susceptible to packet collisions. (vi) Increased node mobility may require more frequent exchange of beacons thus introducing additional overhead and packet collisions. Thus neighbour discovery over two or more hops using beacon messages becomes less reliable.

The simplest NCB mechanisms are "Self Pruning" (Lim and Kim, 2000) and "Neigh-
bour Coverage” (Tseng et al., 2001). Both mechanisms are equivalent. Two neighbour sets are maintained at each node, a set $N_j$ denoting the neighbours of node $j$ and a set $N_i$ denoting the neighbours of node $i$ that node $j$ received the broadcast packet from. When node $j$ receives a broadcast packet for the first time from a node $i$, it determines its coverage set through the set cover calculation shown in Equation 2.1.

$$C_j = N_j - N_i - \{i\} \quad (2.1)$$

The resulting coverage set $C_j$ is the set of neighbours of node $j$ that are not neighbours of node $i$ or not covered by node $i$. This keeps track of pending hosts in $j$’s neighbourhood that have not received a direct broadcast from node $i$ as they are outside node $i$’s broadcast range. Node $j$ is inhibited from rebroadcasting the packet if $C_j$ is an empty set. An empty set implies that all neighbours of node $j$ are also neighbours of node $i$. This calculation is performed by each node that receives a broadcast packet prior to rebroadcasting.

The "Scalable Broadcast Algorithm" (SBA) (Peng and Lu, 2000), utilises two hop neighbour knowledge and a broadcast delay timer to determine whether or not to rebroadcast. A node $j$ upon receiving a broadcast from a node $i$ utilises Equation 2.1 to determine if it has any neighbours that are not reachable from node $i$. If the result is an empty set then the node will not rebroadcast. However, if the result is not an empty set, then node $j$ will schedule a broadcast with a specific delay. The delay may be specified dynamically and is calculated and based upon the current node $j$'s node degree ($D_j$) and its neighbour's maximum node degree ($D_{Nmax}$) as shown by the ratio in Equation 2.2. Equation 2.2 favours nodes with the greatest number of neighbours thus allowing those nodes to broadcast before those nodes with fewer neighbours.

$$T_{delay} = \frac{D_{Nmax}}{D_j} \quad (2.2)$$

"Dominant pruning" as proposed by (Lim and Kim, 2000) makes use of two hop neighbour knowledge and a greedy set cover algorithm to limit the broadcast storm problem. In dominant pruning, unlike previous mechanisms the sender specifies a
set of nodes in a forward list (attached to the broadcast packet) that are responsible for rebroadcasting the packet so that it reaches all nodes within two hops. Finding the minimum forwarding list is a greedy set cover problem that is NP-complete. In (Lou and Wu, 2002), deficiencies of dominant pruning are analysed and two new algorithms, "Total Dominant Pruning" and "Partial Dominant Pruning" are proposed. The algorithms more effectively utilise two hop neighbour knowledge to further reduce redundant broadcasts.

MPR flooding (Qayyum et al., 2001) makes use of two hop neighbour knowledge and is employed in the OLSR routing protocol for the dissemination of link state information. MPR aims to reduce the number of redundant retransmissions during flooding by restricting the number of retransmitters to a small set of neighbouring relay nodes. This set of relay nodes is minimised by efficiently selecting neighbours which provide one hop cover of the network area provided by the complete set of neighbours. These neighbours are the multipoint relays for a given node. As with Dominant Pruning, finding the minimal MPR set is NP-complete. However, the fol-
A lowing efficient heuristic is proposed in (Qayyum et al., 2001) for a node to determine its MPRs:

1. Find all two hop neighbours reachable from only a single one hop neighbour. Assign the one hop neighbours as MPRs.

2. Determine the resultant cover set - the set of two hop neighbours that will receive the packet from the current MPR set.

3. From the remaining one hop neighbours not in the MPR set, find the ones that cover the most two hop neighbours not in the cover set.

4. Repeat from step 2 until all two hop neighbours are covered.

In Figure 2.9, nodes B, D and E are one hop neighbours of node A. Nodes C, G and F are two hop neighbours of node A. A broadcast is initiated by node A. According to the proposed MPR heuristic step 1, node B is selected as a MPR as node C may only be reached by node B. The remaining nodes G and F are similarly covered by node D which is then added to the MPR list. Node E is not added to the MPR list as its neighbouring node F is already covered by node D. In MPR, neighbouring nodes are informed that they are selected as an MPR by a neighbouring node through the attachment of MPR lists to beacon messages.

It is also possible to add the list of MPRs to the broadcast packet itself, thus accounting for the direction of propagation (source based) of the MPR flood as implemented in the Ad Hoc Broadcast Protocol (AHBP) (Peng and Lu, 2002). AHBP selects relay nodes referred to as Broadcast Relay Gateway (BRG) using the same algorithm as MPR. However, given that this information is attached to the broadcast packet, AHBP also performs neighbour elimination using Equation 2.1 to remove any nodes covered by the previous broadcast. AHBP further extends MPR to handle situations where a node \( j \) not having exchanged beacon messages with a broadcasting node \( i \) and not selected as a BRG will assume BRG status upon receiving a broadcast from node \( i \).
The MPR algorithm is source-dependent, requiring that a relay node be aware of the preceding broadcasting node. In (Adjih et al., 2002), a localised algorithm is proposed to make the relay selection source-independent. The algorithm also improves upon MPR by determining a smaller relay set, yet still providing equivalent performance to MPR. In (Wu, 2003), the authors extend this work to further reduce the size of the relay set without introducing additional cost.

In (Wu and Li, 1999) the authors describe a simple and efficient distributed algorithm for determining a connected dominating set (CDS). CDS may be used to limit the broadcast storm problem, by limiting broadcasting nodes to those gateway nodes. A dominating set exists when all nodes in the network either belong to the dominating set or are neighbours of those nodes that belong to the dominating set. The authors define a node $i$ as an "intermediate" node if there exist two neighbours $j$ and $k$ of $i$ that are not direct neighbours of each other. Two rules are also applied:

- **Rule 1** - Given two intermediate neighbouring nodes $u$ and $v$. If neighbours of $u$ are also neighbours of $v$ and the node identifier of node $u$ is less than the node identifier of node $v$ then node $u$ is not an "inter-gateway" node. Therefore node $u$ is covered by node $v$.

- **Rule 2** - Assume three inter-gateway nodes $u$, $v$ and $w$ with shared neighbours. If the neighbours of node $u$ are contained within the neighbours of nodes $v$ and $w$ (that are also neighbours of each other) and node $u$'s identifier is less than both node $v$ and $w$, then node $u$ may be eliminated as a gateway node.

In (Stojmenovic et al., 2002), the authors propose to replace the use of node identifier's as a key in Rule 1 and Rule 2 with a node's neighbour degree and its (x,y) coordinates as additional keys. The neighbour degree is defined as a node's total number of neighbouring nodes. The use of neighbour degree allows for a significant reduction in the resulting size of the dominating set. Nodes that belong to the dominating set are referred to as "internal" nodes. Broadcasting nodes are limited to those nodes selected as internal nodes. Nodes which have unique neighbours as with MPR are always selected as internal nodes.
The mechanisms described so far rely upon explicit reasoning to determine whether or not to rebroadcast. In (Sucec and Marsic, 2000), the Lightweight and Efficient Network-Wide Broadcast (LENWB) mechanism is proposed. LENWB utilises implicit reasoning based upon its knowledge of the reasoning of neighbouring nodes given the knowledge of which nodes received a broadcast packet. LENWB utilises 2-hop neighbour knowledge to determine the node degree of all neighbouring nodes. Each neighbour node is assigned a priority that is proportional to its node degree. A node relies upon its higher priority neighbouring nodes to perform rebroadcasts. Thus LENWB can proactively determine which neighbouring nodes will rebroadcast and also which neighbouring nodes will receive the broadcast. If the node determines that some of its neighbours will not receive a broadcast, then it rebroadcasts the message.

### 2.6.3 Cluster Based (CB) Flooding

Clustering (Gerla and Tsai, 1995) is the process of grouping nodes together into clusters (groups) as shown in Figure 2.10. A representative of each cluster is called the *clusterhead* (nodes B and D). A cluster encompasses all nodes within a clusterheads transmission range. Nodes that belong to a cluster, but are not the clusterhead are called *ordinary nodes*. Often nodes may belong to more than one cluster, these nodes are called *gateways* (node C). Only clusterhead nodes and gateway nodes are responsible for propagating messages. The process of forming clusters may be either active or passive. In Figure 2.10, an ordinary node (A), broadcasts a message to be flooded. The message is received by node B and rebroadcast to all nodes within node B’s broadcast range. Node C being a gateway node receives the message from node B and rebroadcasts the message. The clusterhead Node D receives the message and rebroadcasts it to is neighbouring nodes. The directed solid lines show the propagation of the message among those nodes that are allowed to rebroadcast. Dashed directed lines show the propagation of the message by the clusterheads and the gateway node to ordinary nodes.

In active clustering, nodes must cooperate in order to elect clusterheads. This is achieved through periodic exchange of control information. The formation of clus-
Clusterhead node
Gateway node
Ordinary node

Figure 2.10 Example of a cluster based flood initiated from node A - with only clusterhead and gateway nodes rebroadcasting.

ters in active clustering is independent of the background data traffic. The selection of a clusterhead may be based upon Lowest ID algorithm (Lin and Gerla, 1997) or Highest ID algorithm (Lin and Gerla, 1997). In (Pagani and Rossi, 1997) and (Pagani and Rossi, 1999), clustering is used as an optimised flooding mechanism, whereby only clusterheads and gateways rebroadcast messages. Additionally, the clusterheads in the mechanism ensure reliable delivery of the message to those nodes belonging to their cluster. In (Lou and Wu, 2002) a mechanism that builds a cluster based backbone for the dissemination of information is proposed. They propose the creation of a static and a dynamic backbone. The static backbone is created using a source independent connected dominating set. The dynamic backbone is created using a source dependent connected dominating set.

In passive clustering (Yi et al., 2001) (Yi et al., 2003), cluster formation is dependent on background data traffic. Therefore passive clustering will not form clusters until there is background traffic. This is because, in passive clustering, the flow of data traffic is used to propagate cluster control information and collect neighbour information through promiscuous packet reception. Promiscuous packet reception is
achieved by allowing the MAC layer to pass all received packets up the TCP/IP stack irrespective of MAC address. Passive clustering is beneficial in that it utilises existing data traffic to form clusters. However, without existing data traffic it is unable for form clusters and provide the benefits of an optimised flood. Active clustering requires that cluster control information be exchanged between nodes and clusterheads. Thus it requires more overhead than passive clustering or non-clustered flooding mechanisms for the formation of clusters. However, unlike passive clustering, there is no delay involved as it does not require background traffic.

2.6.4 Power Control Based Flooding

In Power Control Based (PCB) flooding, nodes utilise transmission power control when broadcasting packets. The use of transmission power control allows for the isolation of broadcasts through reduction of transmission range and is beneficial for the following reasons: The required power for a transmission distance of $d$ between two nodes is proportional to $d^\lambda$. Typically $\lambda$ takes a value between 2 and 6, depending on the characteristics of the communications medium (Wisielthier et al., 2000). Isolating a broadcast increases the probability of only necessary nodes hearing a broadcast. This helps to both reduce duplicate packet reception and the power consumed with packet reception at the receiver. Limiting the nodes that will hear a broadcast reduces medium contention between nodes, increases medium utilisation and reduces the probability of packet collisions. The use of transmission power control may result in one high power transmission being replaced with two or more low power transmissions. A common analogy would be, "In a room full of people, it would be better for people to whisper, rather than yell, at one another".

A wireless network may be described by the graph $G = (V,E)$, where $V$ is the set of nodes (vertices) and $E$ the set of edges where $E \subseteq V^2$. Communication between two nodes is possible if an edge $(u,v)$ belongs to $E$. The distance between two nodes $u$ and $v$ is defined as $d(u,v)$. The Relative Neighbourhood Graph (RNG) (Toussaint, 1980) shown in Figure 2.12 is formed when two nodes $u$ and $v$ are connected with an edge, if their lune contains no other nodes of the graph. The lune of two nodes $u$ and $v$, shown in Figure 2.11 (in grey) is defined as the intersection of two spheres of
Figure 2.11 Formation of Relative Neighbourhood Graph using a lune

radius \( d(u,v) \), one centred at node \( u \) and the other at node \( v \). Graphs, such as RNG, in which vertices are connected by an edge, if the edge satisfies some condition of closeness are called proximity graphs.

The use of a distributed RNG with local knowledge was first proposed in (Borbash and Jennings, 2002) as a topology control algorithm to minimise node degrees, hop diameter and maximum transmission range and ensure connectivity. The resulting RNG graph is the same irrespective of if it is calculated in a distributed or centralised manner. In (Cartigny et al., 2002a)(Cartigny et al., 2003a), the authors propose a distributed flooding protocol based upon the RNG called RNG Relay Subset (RRS). RRS allows for self-selection of forwarding neighbours. In RRS a node \( v \) will select itself as a relay for a node \( u \) if and only if node \( v \) is also neighbour of node \( u \). Node \( v \) must also have a RNG neighbour that is not covered by node \( u \)'s broadcast. RRS addresses the broadcast storm problem by reducing the transmission range of a broadcasting node to only include those RNG neighbours that must receive the broadcast, thereby ensuring the flood propagates. The use of self-selection by nodes using RRS allows nodes to determine if they need to rebroadcast without the need for additional information attached to the broadcast packet.
2.6.5 Resource Awareness

There has been very little work in ad hoc network literature for optimised flooding mechanisms that are resource aware. Mechanisms that are resource aware attempt to disseminate information such that they utilise available resources within the ad hoc network in an efficient and aware manner. Thus resource aware flooding mechanisms should select relay nodes based upon their available resources. Available resources may be a node's remaining power reserves or constraints. Constraints are those that are inherently imposed upon a device such as limited broadcast range. Other constraints may be user based constraints, whereby a user limits the participation (benevolence) of a node in network activities based upon the devices remaining battery power.

In Activity Scheduling (Stojmenovic and Wu, 2003), nodes must actively determine if they are in an active or passive state in order that the network remain connected and also the lifetime of both the network and nodes are maximised. Nodes in a passive state (sleeping), do not consume constant energy, they are not effected by reception of packets not destined for them. In (Shaikh et al., 2003), a topology maintenance scheme is proposed with the aim of extending the life time of the network while preserving network connectivity. A node is either active or has a neighbouring node
that is active. Thus, flooding (and routing) activities are restricted to those active nodes. The active nodes create a connected dominating set. Nodes update their activity status periodically during short transition periods when all nodes are active and packets destined for passive nodes are delivered. It is possible for nodes that have significant available energy resources to remain active longer than nodes with less energy, which may enter a passive state more often and on awake to collect packets destined for them. (Shaikh et al., 2003) propose metrics for determining activity status that are based upon combinations of node-degree and remaining battery power.

In (Wu, 2003), the authors extend MPR flooding to reduce the size of the relay set without introducing additional overhead. The process of selecting relays may also be done in a resource aware manner, thus accounting for the remaining battery power of nodes. This mechanism still utilises "step 1" of the MPR algorithm which is to select those nodes with unique neighbours. However, as explained in Chapter 4 and shown in Figure 4.3, the majority of relays that could be selected to relay a message are selected only because they have unique neighbours. Thus the selection of remaining relays based solely upon their resources (battery power) is limited in its results as these relays only constitute a fraction of all relays selected.

2.6.6 A Summary of the Features of Distributed Flooding Mechanisms

Table 2.6.6 provides a summary of the features of distributed and optimised flooding mechanisms discussed in this section. The features of the various mechanisms are described as follows: "Class" specifies the specific class of the flooding algorithm, which may be Heuristic Based (HB), Neighbour Coverage Based (NCB), Cluster Based (CB) or Power Control Based (PCB). "Beacons" specifies whether or not a mechanism requires the use of beacons for exchange of neighbour or relay information. "Neighbour Information (hops)" specifies the number of hops of neighbour information from the source node that is required. "Power Control" specifies whether or not the mechanisms utilises transmission power control to reduce transmission range when broadcasting. "Relay List" specifies whether or not a relay list is attached to the broadcast message. "Reactive / Proactive" specifies how a mechanism coor-
dinates which nodes are responsible for rebroadcasting. "Resource Aware" specifies whether or not the mechanism is able to account for node resources when deciding whether or not to rebroadcast. "GPS" specifies whether global position information is required for rebroadcast decisions. "SS" refers to signal strength measurements and whether a mechanism requires or is able to use such information. "Delay" may be either "jitter" or "timer" based. "Introspection" refers to mechanisms where nodes are able determine whether or not their neighbours will rebroadcast a message.
<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Class</th>
<th>Beacon</th>
<th>Neighbour Info. (hops)</th>
<th>Power Control</th>
<th>Relay List</th>
<th>(R)eactive / (P)roactive</th>
<th>Resource Aware</th>
<th>GPS</th>
<th>SS</th>
<th>Delay</th>
<th>Intr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blind</td>
<td>-</td>
<td>N</td>
<td>N (0)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>HB</td>
<td>N/Y</td>
<td>N/Y (n)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N/Y</td>
<td>N/Y</td>
<td>N/Y</td>
<td>J/T</td>
<td>N</td>
</tr>
<tr>
<td>Location / Dist.</td>
<td>HB</td>
<td>N/Y</td>
<td>N/Y (1)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N</td>
<td>N/Y</td>
<td>N/Y</td>
<td>J/T</td>
<td>N</td>
</tr>
<tr>
<td>S. Pruning</td>
<td>NCB</td>
<td>Y</td>
<td>Y (1)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>SBA</td>
<td>NCB</td>
<td>Y</td>
<td>Y (2)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>T</td>
<td>N</td>
</tr>
<tr>
<td>D. Pruning</td>
<td>NCB</td>
<td>Y</td>
<td>Y (2)</td>
<td>N</td>
<td>Y</td>
<td>R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>MPR</td>
<td>NCB</td>
<td>Y</td>
<td>Y (2)</td>
<td>N</td>
<td>N/Y</td>
<td>R/P</td>
<td>N/Y</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>CDS</td>
<td>NCB</td>
<td>Y</td>
<td>Y (2)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N/Y</td>
<td>N/Y</td>
<td>N/Y</td>
<td>J/T</td>
<td>N/Y</td>
</tr>
<tr>
<td>AHBP</td>
<td>NCB</td>
<td>Y</td>
<td>Y (2)</td>
<td>N</td>
<td>Y</td>
<td>R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>LENWB</td>
<td>NCB</td>
<td>Y</td>
<td>Y (2)</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>Y</td>
</tr>
<tr>
<td>Active Cluster</td>
<td>CB</td>
<td>Y</td>
<td>Y (n)</td>
<td>N</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>Passive Cluster</td>
<td>CB</td>
<td>Traffic</td>
<td>Traffic (2)</td>
<td>N</td>
<td>N</td>
<td>R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>J</td>
<td>N</td>
</tr>
<tr>
<td>RNG</td>
<td>PCB</td>
<td>Y</td>
<td>Y (1)</td>
<td>Y</td>
<td>N</td>
<td>R</td>
<td>N/Y</td>
<td>N/Y</td>
<td>J/T</td>
<td>N/Y</td>
<td>N/Y</td>
</tr>
</tbody>
</table>

Table 2.1 A summary of the features of distributed flooding mechanisms.
2.7 Reliable Flooding in Ad hoc Networks

Flooding is a fundamental mechanism in ad hoc networks. It is therefore important that the reliability of flooding mechanisms be considered. Blind flooding in ad hoc networks is often specified as being a "fall back" mechanism in situations where an optimised flooding mechanism may fail (Basagni et al., 1999). This is due to the inherent redundancy present in blind flooding, where every node will rebroadcast a flooded packet at least once. However, Blind flooding results in the broadcast storm problem. Optimised flooding mechanisms reduce the level of redundancy when performing a flood. This reduction in redundancy limits the broadcast storm problem, but also reduces the reliability and redundancy inherent in Blind flooding. In this section, we review existing literature on flooding mechanisms which address the issue of reliability, however these mechanisms do not address the issues of optimisation with respect to limiting the broadcast storm problem.

In (Alagar and Venkatesan, 1995), a reliable flooding mechanism that builds upon blind flooding is proposed. Each node $i$ maintains a history buffer. The history buffer stores all messages sent and received by a node. A node upon receiving a broadcast message will send an acknowledgement to the broadcasting node. The acknowledgement is sent irrespective of prior reception of the message. A broadcasting node will wait for acknowledgements from all its one hop neighbouring nodes. If no acknowledgement is received then the broadcasting will rebroadcast the message a certain number of times. As with neighbour knowledge based flooding mechanisms, the mechanism utilises beacon messages to discover one hop neighbours. During periods of low network activity, the use of beacon messages is extended to allow nodes to perform a "Handshake Procedure". This allows two nodes to exchange their history buffers and thus determine which messages they need to exchange so that both nodes have heard the same messages. The mechanism is simple, however the use of blind flooding means the mechanism will suffer from the broadcast storm problem. Also, there is a possibility of implosion when nodes send an acknowledgement back to the broadcasting node.

A reliable flooding mechanism for ad hoc networks is described in (Pagani and Rossi,
1997) and (Pagani and Rossi, 1999). The flooding mechanism is based upon the use of a clustering (Gerla and Tsai, 1995) mechanism where nodes are grouped into clusters as described in Cluster Based Flooding (section 2.6.3). Each cluster consists of a single clusterhead that is responsible for nodes within its cluster. Clusterheads are responsible for ensuring messages begin flooded are received by nodes they are responsible for. The clusterhead will wait for acknowledgements from each node within its cluster. The gateway nodes will then forward the message to the clusterheads of other clusters that may also belong to. In this way a message is reliably propagated from cluster to cluster. This mechanism essentially creates a forwarding tree of nodes for the propagation of a flood. The mechanism ensures reliability by utilising unicast messages between cluster heads and the collection of unicast acknowledgements from nodes belonging to a cluster. Gateway nodes will delay acknowledgement of a received message from the preceding clusterhead while they transmit the message to another clusterhead. Given that there may be multiple clusters, this can be seen as being recursive. Once the last cluster is reached then acknowledgements will start flowing back towards the originating clusterhead and ultimately the source of the flood. In this way the source of the flood is able to determine which nodes the flood was received by. The problem with a cluster based approach is the formation and maintenance of the clusters which is costly especially in the presence of mobility. The formation of the cluster tree does not ensure that all nodes are covered by a clusterhead as nodes may leave a cluster. Therefore it is possible for some nodes to be excluded from receiving a broadcast. Additionally given node mobility, the reverse path back to the source node may be destroyed. To solve this problem, nodes may flood acknowledgements back to the cluster head of the originating node.

In (Tourrilhes, 1998), a unique mechanism is proposed whereby a broadcasting node elects a neighbouring node as its "collision detector" in order to detect collisions during broadcast transmissions. The collision detector provides feedback about the success of a broadcast or failure, in which case a rebroadcast occurs. The mechanism relies upon the use of the RTS/CTS mechanism (as found in IEEE 802.11 MAC) and the ability to reserve the broadcast medium through the exchange of RTS/CTS. The mechanism requires some minor modifications to the IEEE 802.11 MAC, whereby prior to broadcasting a frame, the broadcasting node exchanges RTS/CTS with an
elected node (collision detector). This allows the broadcasting node to reserve the medium. The collision detector is able to determine if a collision has occurred within its sphere of influence and inform the broadcasting node of success via an acknowledgement. The problem with this approach is that only one neighbouring node is elected. Given the omni-directional nature of a wireless broadcasts, an elected node may only detect collisions at its own point in space.

In (Tang and Gerla, 2000), the authors propose a simple modification to IEEE 802.11 that utilises a simple RTS/CTS mechanism to enhance delivery of broadcast packets. The mechanism works as follows: A node wishing to broadcast a message first enters the collision avoidance phase of IEEE 802.11. Once this phase is complete, the node then transmits a RTS to its neighbouring nodes and sets a timer to wait for a CTS reply. Neighbouring nodes upon receiving a RTS transmit a CTS and set a timer to wait for a data frame from the source node. The exception to this is when the node is currently in a "YIELD" state, where it may be receiving a transmission from another source. The source node upon receiving a CTS, transmits the data if the medium is free. Nodes that are not involved in receiving the broadcast (ie. not 1 hop neighbours of the source node) and which receive a CTS set their state to "YIELD" so as to not interfere with the broadcast. The authors specify that the mechanism "enhances" broadcasting, as it may improve reliability but not ensure reliable broadcasting due the lack of an acknowledgement. The proposed mechanism does improve upon the standard IEEE 802.11 approach to broadcasting where only CSMA is used. However, the following problem may occur. A source node is allowed to broadcast upon reception a single CTS. The source node may, however, receive multiple CTS frames from neighbouring nodes which may result in contention and collisions. As the nodes transmitting the CTS frames must contend for the medium. This may result in the source node timing out while waiting for a CTS. More importantly as reception of only one CTS is required, nodes that are already in a yield state or receiving a transmission will not reply to the RTS. Thus, the new transmission from the source node may result in corruption of a frame that is currently being received.

In (Stojmenovic et al., 2002), the authors propose RANA (Retransmission After Negative Acknowledgements) broadcasting algorithm. If a node A broadcasts a message
and a collision occurs at node B. Depending on when the collision occurred it may be possible at node B to retrieve the sender identification and message identification from the message. In this case node B will delay a specific period of time before sending a negative acknowledgement message to node A. If during this delay, node B receives the same message from another source, it will cancel the negative acknowledgement to node A. If node A receives the negative acknowledgement without collision, it will then retransmit the message to node B.

In (Hsu and Tseng, 2002), a flooding mechanism is proposed that attempts to limit the broadcast storm problem and also provide reliability. The mechanism consists of 3 phases. The first phase is the scattering phase, in which the source of node initiates a flood that utilises the counter based (Ni et al., 1999) flooding mechanism. The idea is to disseminate the message to as many nodes as possible. A handshake procedure as described in (Alagar and Venkatesan, 1995) is utilised to ensure neighbouring nodes have received the same messages. During the scattering phase a tree graph is formed from all nodes back to the original source of the flood. The second phase is gathering phase, in which acknowledgements are collected from all nodes. Acknowledgements travel back towards the source of the flood via the acyclic graph formed during the scattering phase. Unicast packet transmission is used for the transfer of acknowledgement. The third stage is the purging phase and is initiated by the source node flooding a request for all data structures maintained during the reliable flood at each node to be deleted.

In (Sheu et al., 2002), a reliable flooding mechanism is described. The mechanism consists of two schemes: Duplicate Broadcast Scheme (DBS) and Broadcast Acknowledgement Scheme (BAS). In DBS, a node maintains its local set of 1-hop neighbour nodes in a table called Local Connectivity Table (LCT). When a node broadcasts a message it relies upon the BAS to determine which neighbours received the message. Given the number of successfully received messages and the number of nodes in the LCT, the authors propose to determine whether or not it is necessary to perform an additional broadcast, thus attempting to reach those nodes that did not received the message. The BAS is a positive acknowledgement scheme that involves modifying the IEEE 802.11 MAC, yet maintaining compatibility. The BAS
requires that all nodes successfully receiving a broadcast message respond with an acknowledgement. The scheme allows receiving nodes to utilise the DIFS period after receiving the data frame to transmit an acknowledgement. The DIFS period is divided into mini-slots and nodes select a mini-slot in which to send their acknowledgement to the broadcasting node.

In (Lou and Wu, 2003), the authors propose a simple broadcast algorithm that provides a high delivery ratio for packets being flooded in an ad hoc network and provides limited reduction of redundant broadcasts. The algorithm allows for only selected forward nodes (one hop neighbours) of a broadcasting node to send acknowledgements confirming reception of a broadcast packet. Forward nodes are selected so as to ensure that all two hop neighbours of the broadcasting node are covered. Moreover, no acknowledgment is needed from one hop neighbours, that are covered by at least two forwarding neighbours. The broadcasting node waits for acknowledgements from all of its forwarding one hop neighbour nodes. If not all acknowledgements are received, the broadcast node will rebroadcast the packet until a maximum number of retries is reached.

### 2.8 Information Collection in Ad hoc Networks

Information collection is an all-to-one process, whereby information flows from all nodes in an ad hoc network towards a specific node. Information collection is important in ad hoc networks as it allows for a node to sense the state of the network. Thus, information collection may be used in routing protocols, service discovery, auto-configuration, network management, topology discovery, data retrieval and reliable broadcasting. Little research has been done in information collection mechanisms for ad hoc networks. However, a significant amount of work has been done in a related area of work - sensor networks.

A *Sensor Network* (Estrin et al., 1999) is a specific type of wireless network that allows for a node acting as a sink to perform information collection from one or more nodes within the network. Sensor networks may typically be constructed from thousands of tiny disposable, low power, programmable devices equipped with sens-
ing and wireless communication capabilities. Sensor networks are designed to be deployed near or within the phenomenon that is being sensed. Nodes forming the sensor network may be dispersed randomly. Therefore sensor networks and their associated mechanisms should have the ability to self organise. Besides sensing their environment, nodes are able to perform simple processing (fusion) of phenomenon sensed locally or acquired from neighbouring sensor nodes. Nodes are able to react to requests from sinks for specific types of information as well as being able to advertise the availability of specific information. The ability to perform fusion allows for only processed data to be sent back to a sink rather than the raw data. This significantly increases the capacity and application of sensor networks. Sensor networks have a wide range of applications in modern life. Some application areas may be in environmental monitoring, health care, exploration, disaster recovery, military and security systems.

Sensor networks differ from ad hoc networks in their application, construction, characteristics and constraints in the following ways (Akyildiz et al., 2002):

- The number of sensor nodes in a sensor network may be several orders of magnitude higher than in an ad hoc network.

- Sensor nodes are more densely deployed than ad hoc network nodes.

- Sensor nodes are prone to failures.

- Sensor nodes mainly use a broadcast communication paradigm, whereas most ad hoc networks are based on point-to-point communications.

- Sensor nodes are more severely limited in power, computational capabilities and memory than ad hoc network nodes.

- Possible lack of global identification because of the associated overheads and the large number of sensors.

- Minimal or non-existent user interaction.
The main priority of sensor networks is for the flow of information from sensors back to the sink - information collection. However, in an ad hoc network, information collection may be of secondary importance to point to point communication. Information collection in an ad hoc network must be performed in such a way as to have low overhead so as to not affect the normal operation of the ad hoc network. For these reasons, protocols suitable to ad hoc networks are not necessarily suitable to sensor networks and vice versa. In ad hoc network literature there has been a strong focus on information dissemination protocols for routing (one-to-one) and flooding (one-to-all), but little on information collection (all-to-one) mechanisms in ad hoc networks.

The process of information collection may be seen as "data centric", whereby mechanisms may be either sink oriented or source oriented. In the sink oriented approach, a sink node initiates a request for information by flooding the network. The request specifies the type of information the sink is requesting. Nodes or sensors receiving the request may then reply to the request if they have the requested information. In the source oriented approach, source nodes advertise availability of specific information by flooding the network with an advertisement describing the available information. A sink node may then reply to the advertisement requesting the specific information be sent to it by the source.

Although protocols designed for sensor networks may not be fully applicable to ad hoc networks, there is existing literature in the area that may be applied to the design of ad hoc network sensing protocols. In this thesis we only focus on those mechanisms that are relevant to information collection in IEEE 802.11 based ad hoc networks.

Blind flooding is a simple mechanism that may be used for information dissemination in ad hoc networks and for information collection in sensor networks. However, Blind flooding is a one-to-all mechanism and along with the broadcast nature of wireless communication it introduces significant problems (Akyildiz et al., 2002)(Tseng et al., 2001), such as:

- **Broadcast storm problem:** As flooding may be used to disseminate informa-
tion, sensor networks are susceptible to the negative effects of flooding which include redundant broadcasts, medium contention and packet collisions.

- **Implosion**: Implosion is a situation where a node will receive duplicate packets from a source due to multipath propagation.

- **Resource blindness**: Flooding does not account for the characteristics, constraints and state of devices. Nodes low in battery power or heavily loaded will therefore receive and relay packets when other nodes may be more suitable.

- **Overlap**: Overlap implies that nodes may share common information, which if disseminated may introduce unnecessary and additional overhead.

Gossipping (Pelt, 1996) is another dissemination mechanism used in ad hoc networks that may also be used in sensor networks. In gossipping a node wishing to transmit information randomly selects a single neighbour and unicasts the message to that neighbour. Each receiving neighbour repeats the same process. In this way the data is disseminated throughout the ad hoc network. Gossipping avoids the broadcast storm problem and implosion problems associated with blind flooding as there is only one copy of the message being transmitted, unlike flooding where there may be multiple copies. Therefore gossipping is able to significantly reduce the power consumed during dissemination. However, it also results in the flood progressing at a significantly slower pace when compared to flooding.

In (Chandra et al., 2002), an information collection mechanism for topology discovery in ad hoc networks is described. The mechanism consists of a diffusion phase and a gathering phase. In the diffusion phase, a topology request message is flooded. Upon receiving the topology request, each node marks the preceding broadcasting node as its parent, notifies the parent of the relationship and rebroadcasts the message. At the completion of the diffusion phase, each node has broadcast the message at least once. The result is a network-wide spanning tree with the originating node as the root of the tree. The second phase is initiated by the leaf nodes in the spanning tree replying to the topology request and progresses back towards the root node. A node in the spanning tree waits for all its child nodes to send it a reply before
sending a reply itself to its parent node - thus performing data aggregation. The root node is then able to construct the network topology from all received replies. To ensure reliable delivery of the request in the diffusion phase, a broadcasting node will rebroadcast the request a certain number of times until it receives all acknowledgements from its neighbouring nodes. To ensure adaptability, nodes select multiple parents thus forming a mesh with alternative routes back towards the root node.

Sensor Protocols for Information via Negotiation (SPIN) (Heinzelman et al., 1999) is a set of resource adaptive protocols that attempt to address the deficiencies of flooding through local negotiation and resource adaption. The SPIN protocol is source oriented and builds upon a simple idea that the performance of a sensor network may be improved by using mechanisms that advertise data by sending a concise description (referred to as meta-data) instead of the data itself. When a node receives new data it broadcasts an advertisement containing the meta-data to its local neighbours. These neighbours may then check the meta-data to determine if it has already been received. If not, then a request for the data is sent to the source node. SPIN reduces power consumption by eliminating requests for redundant transmission of data. Nodes are also resource adaptive as they make informed decisions about disseminating information and also monitor their remaining energy levels.

Low Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman et al., 2000) attempts in a distributed manner to minimise energy dissipation by randomly selecting sensor nodes as cluster heads so as to spread the high energy cost of communicating with a base station to all nodes in the network. LEACH is a two phase protocol - setup and steady phase. In the setup phase all nodes choose a random number and compare this to a threshold value. If less than a threshold the node is a clusterhead. The new clusterheads then advertise their status to the entire network by flooding a message. Sensor nodes attach themselves to a clusterhead based upon signal strength and inform the clusterhead of their attachment. Sensor nodes are allotted time to send data to a clusterhead based upon a TDMA approach. In the steady phase sensory nodes may send data to clusterheads which may aggregate all data received from sensory nodes before sending this data to the base station. After a period of time the network enters the setup phase again.
Directed Diffusion (Intanagonwiwat et al., 2000) is a data centric approach to information collection. In this approach a sink initiates a broadcast and attaches its "interest" - a query describing the information of interest to the sink. Each sensor node then stores this interest in a cache along with a time-stamp and gradient fields. As the interest is propagated (using blind flood), nodes setup reverse gradient paths from all nodes to the sink, in a distributed manner. A sensor will send new data of interest back to the sink via the gradient path. Data may be aggregated at intermediate nodes. When a sink begins to receive data of interest, it must repeatedly re-broadcast interests in order to refresh and reinforce the gradients from the sources. Directed Diffusion positively re-enforces certain paths and negatively others in order to remove paths that have failed nodes in them, thus it is able to react to changes in node conditions. This type of information collection is not particular suited for single queries, but rather for persistent queries where the sink is expecting to receive information over a period of time.

In (Sohrabi et al., 2000), Sequential Assignment Routing (SAR), a set of algorithms that perform organisation, management and mobility management are proposed. SAR generates multiple trees where the root of each tree is a one hop neighbour from the sink. Each tree grows outward from the sink. Nodes with low resources and low energy reserves are avoided when forming the trees. Nodes may belong to more than one tree, which allows nodes to chose a tree to relay sensory information back to the sink depending on a tree’s additive QoS metric and energy resources.

In Geographical and Energy Aware Routing (GEAR) (Yu et al., 2001). A sink specifies a target region for each query packet to be sent to. A set of heuristics that account for geographical positions are used to route packets towards the target region. Once the packet reaches the target region, a recursive geographic forwarding mechanism is used to disseminate the query. Each node within the target region divides its local region into sub-regions and sends the query to each of its sub-regions. This recursive mechanism terminates when a node is the only node within a region.

In (Deb et al., 2002), a topology discovery algorithm (TopDisc) for network management in wireless sensor networks is proposed. TopDisc discovers a set of "distinguished nodes" that contain local neighbourhood information. TopDisc logically
organises the network into a tree of clusters (TreC) with distinguished nodes forming the clusterheads. The TreC is rooted at the sink or node requesting topology information. Clusterheads have local knowledge of the network and are responsible for replying back to any topology request with their local knowledge. Limiting replies to clusterheads greatly reduces the communication overhead. The TopDisc algorithm for determining clusterheads is a greedy log(n) approximation algorithm for set coverage. At each stage a node is selected from a the set of discovered nodes that should cover the maximum remaining undiscovered nodes. The reason for this is that nodes have no knowledge of their surrounding neighbours or link states at the start of topology discovery. As the topology discovery request is initiated from the sink and propagated using blind flooding, every node will rebroadcast the packet once. Therefore, neighbour knowledge is generated, "on the fly", as the topology request propagates throughout the network. To assign nodes a role such as being a clusterhead, a node colouring approach is used. The colour a node is assigned is dependent upon the number of topology discovery messages it has heard. The authors describe two different node colouring approaches based on three and four colours. TopDisc is ideal in sensor networks as nodes need only broadcast once during a blind flood. The result is a self organised clustered tree of nodes with the sink as the source. In sensor networks this is important as nodes do not exchange beacon messages as in ad hoc networks. Replies are transmitted back towards the sink via clusterheads thus allowing for efficient data collection through data aggregation.

2.9 Ad hoc Network Simulation

The inherent nature of ad hoc networks as described earlier in Section 2.3 makes mathematical modelling and physical implementation of proposed mechanisms extremely complicated and time consuming. As ad hoc networks are comprised of multiple entities (nodes) interacting in a complex and non deterministic manner, it is not sufficient to only model a single entity nor feasible to model all the interactions that occur. This is further complicated by the equally complex interaction of various protocols that operate from the physical layer (IEEE 802.11) up through the TCP/IP protocol stack. Thus, the use of simulation in ad hoc network research has proven
indispensable and necessary in gathering an understanding of the interactions and performance of proposed mechanisms in ad hoc network research.

Simulation tools allow researchers to gather an understanding of the complex interactions and resulting performance achieved by proposed mechanisms in an environment that allows for repeatability of experiments and easy prototyping of proposed mechanisms. There exist a significant number of commercial and non-commercial simulation tools that allow testing of proposed ad hoc network research - in the case of this thesis information dissemination and information collection in ad hoc network environments. The most popular commercial simulation tools are QualNet (SNT, 2004) and OPNET (OPNET, 2004), while the most popular non-commercial simulation tools are NS-2 (NS-2, 2004) and a non-commercial version of QualNet called GlomoSim (Bajaj et al., 1999).

In this thesis, we decided that it was important to not only understand the performance of proposed mechanisms in realistic wireless ad hoc simulation environments, but to abstract from the underlying complexity in order to analyse and implement proposed mechanisms such that their results and implementation are not specific to a particular physical layer (such as IEEE 802.11) or protocol stack (TCP/IP) and may therefore be applied to future technology. This is the same approach adopted in (Li and Nikolaidis, 2001)(Stojmenovic et al., 2002)(Adjih et al., 2002)(Wu, 2003).

An event based simulation environment was developed in C++ to analyse the performance of our proposed mechanisms and existing published mechanisms. In the simulation, time is divided into epochs. An ideal MAC layer is assumed, such that there is no medium contention nor hidden-node scenarios as it is assumed that during an epoch all nodes may complete transmission of packets without collision or error. All nodes within the selected transmission range of a node will receive a broadcast packet, thus reachability is always 100 percent. This allows us to more accurately determine the actual behaviour and performance of optimised flooding mechanisms at limiting the Broadcast Storm Problem. The effects of a more realistic propagation environment, IEEE 802.11 MAC layer and background traffic are examined in Chapter 5.
To further test proposed approaches in non-ideal environments we chose to use the GlomoSim simulation environment. GlomoSim is a parallel simulation environment implemented in PARSEC, PARallel Simulation Environment for Complex Systems (UCLA, 2004). GlomoSim provides a layered structure that includes radio propagation, medium access control (MAC), link layer, network layer, transport layer and application layer implementations. In addition, various wireless and ad hoc network protocols are already implemented. GlomoSim allows for detailed modelling of several layers and the study of their interaction, while preserving very good runtime efficiency. GlomoSim forms the basis of experiments performed in Chapter 5, where experimentation requires the combined interaction of radio propagation and MAC layers to further understand the performance of optimised flooding mechanisms in non-ideal ad hoc networks where packet loss and network load become important factors.

Given that wireless mobile devices are constrained by limited battery power, the issue of energy consumption in ad hoc networks is fundamental to the understanding of the performance of both information dissemination and information collection mechanisms. It is therefore important to utilise a common energy model for determining energy consumption due to transmission and reception of packets during simulation. All simulations in this thesis assume that the wireless transceivers have power control and thus consume the minimal required energy to reach the intended recipients. All simulations assume an energy model based upon the first order radio model as defined in (Heinzelman et al., 2000) when calculating the cost of transmitting and receiving packets. The first order radio model is further described in Chapter 3.

2.10 Summary

In this chapter we have focused upon mechanisms for information dissemination and information collection in ad hoc networks. The problems associated with both information dissemination and information collection have been introduced and the relevant mechanisms in literature discussed with an aim to highlight the open research areas that are addressed in this thesis.
2.10.1 Summary of Open Research Issues Identified in Current Information Dissemination Literature

There exists significant research in optimised flooding mechanisms that aim to reduce the broadcast storm problem. Optimisation is achieved by reducing the number of redundant broadcasts compared to blind flooding. Heuristic based approaches make finely controlled decisions based on knowledge of the environment, but do not ensure performance or delivery. Neighbour coverage and cluster based approaches limit the number of nodes rebroadcasting through the selection of specific nodes responsible for rebroadcasting messages. These nodes form a connected dominating set thus ensuring delivery in an error free environment. The use of transmission power control as a means of reducing the broadcast storm problem was not considered until Neighbour Aware Adaptive Power Flooding (Chapter 3) and the use of RNG in (Cartigny et al., 2002a). The use of transmission power control in flooding is significant as it allows for the effect of a broadcast to be limited to those nodes that the broadcast is intended for. Limiting transmission power also reduces power consumption associated with transmitting and receiving packets.

Given the heterogeneous nature of ad hoc networks, devices will have varying available resources and be subject to varying (user based) constraints imposed. A device's remaining battery power is an example of an available resource that is limited and may lead to constraints being imposed upon the behaviour of the device by a user. In the reviewed literature there is little work on making optimised flooding mechanisms "resource aware". We see this as an important element (as discussed in Chapter 4) in that optimised flooding mechanism's must account for available resources and constraints in order to effectively disseminate information. Selecting a node as a relay irrespective of its resources or constraints may result in the flood not propagating. Additionally, given varying degrees of mobility some nodes may continuously be selected by an optimised flooding mechanism as they provide optimal coverage. However, this may have the effect of depleting the available power source of these nodes.

To date, literature on flooding mechanisms has not considered the performance of these mechanisms in the presence of background traffic. Thus flooding mechanisms
in an error free environment may have a 100% delivery ratio. However, in an error prone environment with background traffic (this may be TCP or UDP traffic generated by users) these flooding mechanisms may experience problems. Blind flooding has a high degree of redundancy as all nodes will rebroadcast the message at least once. However, the process of optimising a flood by reducing the number of redundant broadcasts also reduces the redundancy and thus the survivability of the flood. Optimised flooding mechanisms utilise unreliable broadcasting as a means of delivering a packet to all neighbouring nodes. Given that broadcasting in IEEE 802.11 only uses carrier sensing before broadcasting results in increased probability of packet collisions. Thus with background traffic, optimised flooding mechanisms are particularly susceptible to failure. The use of a reliable mechanism such as unicasting or reliable broadcast at the MAC layer that provides acknowledgement of received packets would result in a significant improvement. However, with unicast transmission, the packet being flooded must be transmitted to each node that the broadcasting node is responsible for. Existing reliable flooding mechanisms still rely upon blind flooding for dissemination and are therefore susceptible to the broadcast storm problem. The exception is the cluster based method described in (Pagani and Rossi, 1999), where the clusterhead is responsible for ensuring broadcasts are received by nodes within their cluster. Thus in this thesis we propose a reliable and optimised flooding mechanism (Chapter 5) that utilises a combination of the minimum spanning tree and unicast transmission. The minimum spanning tree is used in a similar manner to LMSTFlood to limit the number of nodes a relay is responsible for rebroadcasting a packet to. The reduced number of nodes allows broadcast transmission (which are unreliable) to be replaced by unicast transmission, which provides more reliable packet delivery due to link layer acknowledgement and retransmission.

2.10.2 Summary of Open Research Issues Identified in Current Information Collection Literature

Significant literature exists on mechanisms for information collection in sensor networks. However, sensor networks differ from ad hoc networks in their application, construction, characteristics and constraints. The main priority of sensor networks is for the flow of information from sensors back to the sink. However, in ad hoc
networks there is no emphasis on the direction of flow of information given the user oriented point to point communication. Thus, it is important that more research be performed into information collection in ad hoc network mechanism. Information collection is important in ad hoc networks because it enables various applications such as resource and service discovery, topology discovery, auto configuration, multicast and reliable flooding. Node’s in ad hoc networks maintain information about their neighbouring nodes and possibly more given the use of routing protocols. Information collection in ad hoc networks can occur in two phases: setup and capture. Setup phase involves disseminating a request for information and initiating a backbone that allows for the flow of information back to the source (sink) of the request. The Capture phase involves nodes responding to requests for information and intermediate nodes utilising the backbone created during the setup phase. Thus issues of reliability, resource awareness, optimisation and recovery (in the presence of mobility) exist. In Chapter 6 we have proposed an optimised resource aware information collection mechanism that attempts to solve some of these issues.
Chapter 3

Optimised Information Dissemination

3.1 Introduction

Mechanisms for information dissemination (flooding) form the basis of communication in ad hoc networks. However, flooding in ad hoc networks is problematic as it results in the broadcast storm problem. The broadcast storm problem states that flooding in ad hoc networks suffers from redundant broadcasts, medium contention and packet collisions.

In this chapter we introduce two new optimised flooding mechanisms: Neighbour Aware Adaptive Power (NAAP) flooding and Localised Minimum Spanning Tree flooding (LMSTFlood). These optimised flooding mechanisms are designed to reduce the broadcast storm problem through the use of transmission power control when broadcasting. Existing optimised flooding mechanisms, as described in Chapter 2, do not utilise transmission power control. Instead they rely upon nodes having a constant transmission range when broadcasting. The exception is the Relative Neighbourhood Graph (RNG) (Toussaint, 1980) based mechanism, RNG Relay Subset (RRS) (Cartigny et al., 2002b) which was published independently and concurrently with NAAP.

The use of transmission power control and the associated reduction in transmission range is important for the following reasons: (i) Reducing transmission range results in reduced energy consumption in the transceiver’s amplifier. The reduction in
energy consumption is dependent upon the optimised flooding mechanism’s ability to select a reduced set of closest neighbouring nodes to include within a broadcast. (ii) Isolation of broadcasts through reduced transmission range results in only those necessary nodes receiving the broadcast. Thus nodes experience less received and duplicate packets. Reception of packets is costly (Heinzelman et al., 2000) as each received packet must be processed by the transceiver’s electronics and MAC layer, thus the fewer packets received the less power consumed. (iii) Reduced transmission range limits the effect of a broadcast, thus medium contention and packet collisions are reduced thereby increasing the capacity of the network. The amount of reduction is dependent upon the degree to which the broadcast is localised to as few nodes as possible. (iv) Reducing nodes affected by broadcasts to only those necessary nodes (where possible), improves the scaleability of the optimised flooding mechanism.

In order to adapt transmission power during broadcasting and still ensure continuation of a flood requires that an optimised flooding mechanism be able to: (i) select a reduced set of relays (ii) select relays that are both closest to the broadcasting node and provide coverage of those neighbouring nodes not included within the broadcast, in order for the continuation of the flood. (iii) eliminate nodes that may have received the previous broadcast from another relay using a different transmission power.

Both NAAP and LMSTFlood utilise different techniques for determining a broadcast set. NAAP utilises a combination of techniques based on neighbour coverage, knowledge of neighbouring relays and the decomposition of high power broadcasts into low power broadcasts through local optimisation to reduce transmission range. LMSTFlood utilises the minimum spanning tree (MST) in distributed manner and thus benefits from the properties of the distributed MST and the ability for relay nodes to be self selecting, thus reducing the per packet overhead found in NAAP and other optimised flooding mechanisms. These two different approaches exhibit different properties making them suitable to different applications.

In this chapter we compare the performance of NAAP and LMSTFlood to existing flooding mechanisms. We show that the use of transmission power control allows for significant performance improvement over an optimised flooding mechanism (Multi-
Optimised Information Dissemination

point Relay flooding) that does not utilise transmission power control. We also look at the resulting inherent characteristics of the proposed flooding mechanisms and how these affect their use in ad hoc network applications.

3.2 The Cost of Packet Transmission and Reception

The simulations in this thesis assume that the transceivers have power control and thus consume the minimal required energy to reach the intended recipients. We assume a first order radio model as defined in (Heinzelman et al., 2000) to calculate the cost of transmitting and receiving packets. In this model the first order radio dissipates $E_{elec} = 50nJ/\text{bit}$ to run the circuitry of a transmitter or receiver and a further $E_{amp} = 100pJ/\left(\text{bit} \times m^2\right)$ for the transmitter's amplifier. Equation 3.1 is used to calculate the cost associated with transmitting a $k$-bit message a distance $d$. Equation 3.2 is used to calculate the cost associated with receiving a $k$-bit message. The radios have power control and consume the minimal required energy to reach the intended recipients.

$$E_{TX}(k, d) = E_{elec} * k + E_{amp} * k * d^\lambda$$

(3.1)

$$E_{Rx}(k) = E_{elec} * k$$

(3.2)

The required power for a transmission distance of $d$ between two nodes is proportional to $d^\lambda$. Typically $\lambda$ takes a value between 2 and 6, depending on the characteristics of the communications medium (Wisielthier et al., 2000). In this thesis we make the assumption that the power consumed by a transmitter's amplifier circuitry for packet transmission is proportional to the square for the distance ($\lambda = 2$) and the packet size ($k$) (Heinzelman et al., 2000).
3.3 Neighbour Aware Adaptive Power Flooding

NAAP is a neighbour coverage based optimised flooding mechanism that utilises transmission power control, neighbour elimination, neighbour awareness and local optimisation techniques to limit the broadcast storm problem. NAAP utilises local neighbour knowledge of up to two hops obtained through the exchange of beacon messages. The mechanism is distributed as decisions about rebroadcasting are made by each node.

The main technique used by NAAP is transmission power control. As explained earlier, transmission power control has the potential to greatly impact the performance of an optimised flooding mechanism at limiting the broadcast storm problem as the node density increases. However, the use of varying transmission powers by broadcasting nodes introduces problems in traditional neighbour coverage based flooding mechanisms. These mechanisms rely upon the ability to eliminate neighbours based upon nodes broadcasting with a set transmission power. However, this is not possible if nodes utilise varying transmission powers. NAAP solves this problem by utilising a combination of neighbour elimination and neighbour awareness.

NAAP utilises beacon messages not only to exchange basic topology information, but also to exchange GPS information of neighbouring nodes or signal strength measurements between neighbouring nodes, depending on which is available. This extra information is used by a node receiving a broadcast to determine which of its neighbouring one hop and two hop neighbours may also have received the message. In this way nodes are able to perform neighbour elimination when determining whether or not to rebroadcast.

An optimised flood using NAAP is shown in Figure 3.1. The flood is initiated by node 1. Relays are shown as solid black nodes, non-relay nodes are shaded. Nodes 4 and 5 are elected as relays by node 1. Node 6 is elected as a relay by node 5. In the figure the dashed white arrows represent broadcasts from relay nodes 4 and 5 without neighbour awareness. The dashed broadcast circles are full power broadcasts and the shaded circles are power controlled broadcasts.
A unique mechanism in NAAP is *neighbour awareness*, which implies that neighbouring relay nodes are aware of each other (given that relay information is attached to the broadcast message) and of their shared 1-hop neighbours. As seen in Figure 3.1 the neighbouring relays (nodes 4 and 5) of node 1, will only consider their shared nodes (nodes 9 and 18) that they are closest to, when determining which neighbouring nodes to rebroadcast to. Node 4 is closest to node 9 and node 5 is closest to node 18. Thus they select their closest neighbours. This helps to limit the range of nodes to which a relay is responsible for rebroadcasting to and helps to reduce broadcast overlap. Prior to broadcasting a message, NAAP attaches an optimised relay set to each broadcast message. This set is referred to as $O_i$ and contains both the current relay set and any previous relay nodes that are neighbours of the current relay. Using neighbour awareness, a broadcasting node is able to determine if any of the nodes it is responsible for rebroadcasting to may have already heard a message as they are closer to a previous relay. Thus reducing the problem associated with neighbour elimination when using varying transmission powers.
NAAP implements a simple *local optimisation* mechanism to decompose a high power transmission into two or more low power transmissions as seen in Figure 3.1. In the figure, node 5 is able to reach nodes 14 and 15 with a high power transmission, but decides to decompose the transmission into two smaller transmissions thus selecting node 6 as a relay, which will then cover node 14 and node 15 with a low power transmission. The decision for this decomposition is based upon the difference in equivalent distance between nodes 6, 14 and 15 being greater than a selected threshold distance. Thus node 5 need only transmit to node 6 and node 18. This introduces an additional hop and further allows for the benefits of power control to be achieved. However, it also introduces additional delay as the broadcast must be processed at each node. Given the first order radio model (Heinzelman et al., 2000) used in the simulation, there is a cost associated with receiving and transmitting packets. This cost does not only concern the propagation of the signal but also the cost of powering receiver and transmitter electronics as well as processing requirements. Therefore it may be necessary to limit the decomposition of a broadcast as introducing excessive hops may have negative effects as the energy saved due to reduced transmission distance is negated by the energy consumed for multiple transmissions and receptions. Additionally, introducing excessive hops limits the use the flooding mechanism for route discovery as shown later. The minimum distance for decomposition of broadcasts is controlled in NAAP by $\text{RangeLimit}$. The $\text{RangeLimit}$ specifies the minimal distance below which decomposition will not occur. Thus a small $\text{RangeLimit}$ will have the effect of introducing more low power transmissions. A large $\text{RangeLimit}$ will ensure fewer low power transmissions. In Section 3.7, we provide a comparison of the performance of NAAP with varying $\text{RangeLimits}$.

To summarise the above we give the following intuitive explanation of NAAP using Figure 3.1, where: $u = 1$, $i = \{4, 5\}$ and $j = \{6, 8, 9, 14, 15, 18\}$:

1. Upon receiving a broadcast message(s) from a broadcasting node $u$, each node in $i$ (selected by $u$ as a relay) determines which of its one-hop neighbours also received the same message.

2. Each relay in $i$ determines its remaining neighbours, which did not receive a
message (based upon its knowledge). Each relay then determines its closest set of nodes from \( j \) compared with other neighbouring relays in \( i \) and allocates these nodes to its optimised relay set that it is responsible for rebroadcasting to.

3. If nodes in the resulting optimised relay set are not of an equivalent distance from the relay, it may perform a local optimisation on the set to select a minimal subset of relays that will ensure delivery to remaining nodes in the original optimised set. Otherwise the relay determines a transmission range equal to that of the farthest neighbour it is responsible for.
3.3.1 NAAP Implementation

To implement NAAP, we have made use of neighbour set calculations. The three algorithms \textit{naap()}, \textit{localOptimise}(O_j) and \textit{PM}(m, A, R) implement the NAAP mechanisms. A source node initiating a flood in NAAP must first determine an initial optimised coverage set of nodes using the algorithm \textit{localOptimise}(O_j).

In NAAP, prior to broadcasting a packet, the broadcast transmission power is adjusted to the maximum required transmission power to reach those necessary nodes in the optimised coverage set. The optimised coverage set is then attached to the broadcast packet. Relays upon receiving a broadcast packet from a node \( i \) determine an optimised coverage set \( O_j \) based upon the previous optimised coverage set by using the algorithm \textit{naap>().

The algorithm \textit{PM}(m, A, R) is used to compare the distance (or required transmission power) to the set of nodes in \( A \) from a broadcasting node \( m \) and the set of relay nodes in \( R \). If a node in \( A \) is closer to node \( m \) than it is to one of the relay nodes in \( R \), then the node from set \( A \) is added to a resultant set. Thus a node \( m \) is able to select its closest neighbouring nodes in \( A \), which it may share with other relays that are in the set \( R \).

\begin{algorithm}
Result ← \emptyset
\begin{algorithmic}
1. \textbf{for} all \( x \in A \)
2. \hspace{1em} \textbf{for} all \( n \in R \)
3. \hspace{2em} \textbf{if} (distance(m, x) < distance(n, x))
4. \hspace{3em} Result ← Result + x
5. \hspace{2em} \textbf{endif}
6. \hspace{1em} \textbf{endfor}
7. \textbf{endfor}
8. \textbf{return} Result
\end{algorithmic}
\end{algorithm}

Algorithm \textit{localOptimise}(O_j) attempts to determine a reduced set of closer neighbours of node \( j \) that provide complete coverage of \( j \)'s further neighbours through the
introduction of an additional broadcast. This decomposes a large broadcast trans-
mission from node \( j \) to all its neighbours into smaller transmissions from the reduced
set of closer neighbours. Transmission power consumption is reduced and the area
effected by broadcasting is minimised thereby allowing for an increase in channel
utilisation. It is possible to make use of other local optimisation algorithms which
may rely on more intelligent mechanisms to allocate the minimal set of next hop
neighbours.

**Algorithm**  \( \text{localOptimise}(O_j) \)

1. \( \text{Range} \leftarrow \text{MaxTransmissionDistance} * \text{RangeLimit} \)
2. \( \text{TempO}_j \leftarrow O_j \)
3. \( O_j \leftarrow \emptyset \)
4. \( \text{while } \text{TempO}_j \neq \emptyset \)
5. \( n \leftarrow \text{minimumDistanceNode}(\text{TempO}_j) \)
6. \( \text{if } \text{Distance}(n) > \text{Range} \)
7. \( \text{TempO}_j \leftarrow \text{PM}(j, \text{TempO}_j, n) \)
8. \( \text{endif} \)
9. \( \text{TempO}_j \leftarrow \text{TempO}_j - n \)
10. \( O_j \leftarrow O_j + n \)
11. \( \text{endwhile} \)
12. \( \text{return } O_j \)

In algorithm \( \text{localOptimise}(O_j) \) at line 7, algorithm \( \text{PM}(m, A, R) \) is used to
determine an optimal set of relays where nodes that are closer to a chosen relay are
removed from the optimised set \( \text{TempO}_j \), effectively allocating those nodes to the
chosen relay. This process is repeated until all nodes are allocated to a chosen relay.

In algorithm \( \text{naap()} \) at line 5, the set calculation \( O_j = N_j - O_i - \text{heard}_i - \{i\} \) is used
to determine a set of neighbouring nodes that are unique to node \( j \) and not the source,
node \( i \), of the broadcast. Each node upon receiving a broadcast also determines which
neighbour nodes would have heard the broadcast. This may not include any previous
optimised nodes.
In algorithm \textit{naap()} at line 8, algorithm $PM(m, A, R)$ is used to remove nodes from node $j$'s broadcast set ($O_j$) that may overlap with neighbouring relays in $O_i$. Those nodes that are closer to neighbouring relays in $O_i$ (than they are to node $j$) are removed from $O_j$. Thus neighbouring relays in $O_i$ are responsible for those shared neighbouring nodes to which they have the lowest required transmission power and lose those neighbouring nodes to which they have a greater required transmission power.

\textbf{Algorithm \textit{naap}()}

1. $i \leftarrow$ previous broadcasting relay
2. $O_i \leftarrow$ previous optimised set attached to broadcast packet
3. $j \leftarrow$ current broadcasting relay
4. $\text{heard}_i \leftarrow$ nodes that may have heard previous broadcast
5. $O_j \leftarrow N_j - O_i - \text{heard}_i - \{i\}$
6. $\text{NeighbouringRelays} \leftarrow N(O_i \cap N_j)$
7. \textbf{if} $O_j \neq \emptyset$
8. \hspace{1em} $O_j \leftarrow PM(j, O_j, \text{NeighbouringRelays})$
9. \hspace{1em} \textbf{if} nodes in $O_j$ do not have equivalent required transmission power
10. \hspace{2em} $O_j \leftarrow \text{localOptimise}(O_j)$
11. \hspace{1em} \textbf{endif}
12. \hspace{1em} $\text{MaxTransmissionPower} \leftarrow \text{maximumPower}(O_j)$
13. \hspace{1em} $O_j \leftarrow O_j + (O_i \cap N_j)$
14. \hspace{1em} \textbf{endif}

At line 9 in \textit{naap()}, we determine if all nodes in $O_j$ have an equivalent required transmission power within a small tolerance. If not, this implies that some nodes are closer to node $j$ and others further away. At line 12 in \textit{naap()}, the required transmission power is adjusted to reach those necessary relay nodes in $O_j$, which will be responsible for rebroadcasting. At line 13, any neighbouring relays from $O_i$ that were selected by the previous broadcasting node are attached to the optimised set $O_j$. This allows the next set of relays to determine if any of their neighbouring nodes have been covered by prior relays and aids in reducing the effects of varying transmission powers used by prior relays.
3.3.2 NAAP Worked Example

In this section, we provide a worked example of NAAP, MPR and Blind flooding. Figure 3.2 shows the result of a Blind flood from a source node 1, with all nodes performing full power packet transmissions. Figure 3.2 allows the set of neighbours for each node to be determined as follows:

\[ N(1) = \{2, 3, 4, 5\}, \quad N(2) = \{1, 16\}, \quad N(3) = \{1, 7\}, \quad N(4) = \{1, 5, 8, 9, 18\}, \]

\[ N(5) = \{1, 4, 6, 9, 14, 15, 18\}, \quad N(6) = \{5, 14, 15, 18\}, \quad N(7) = \{3, 17\}, \]

\[ N(8) = \{4, 9, 10, 18\}, \quad N(9) = \{4, 5, 8, 18\}, \quad N(10) = \{8, 11\}, \quad N(11) = \{10, 12, 13\}, \]
Figure 3.3 An example of Multipoint Relay flooding initiated from a source node (node 1).

Figure 3.4 An example of Neighbour Aware Adaptive Power flooding initiated from a source node (node 1).
Optimised Information Dissemination

\[ \begin{align*}
N(12) &= \{11\}, N(13) = \{11\}, N(14) = \{5, 6, 15\}, N(15) = \{5, 6, 14\}, \\
N(16) &= \{2\}, N(17) = \{7\}, N(18) = \{4, 5, 6, 8, 9\}
\end{align*} \]

Using NAAP, node 1 initiates a broadcast and determines \( O_1 = \{2, 3, 4, 5\} \). \( O_1 \), which is then attached to packet \( P \) and broadcast at the maximum required transmission power given the nodes in \( O_1 \). \( P \) is received by nodes 2, 3, 4 and 5. We ignore nodes 2 and 3 and concentrate on flooding at nodes 4 and 5 as this is the more complex area of the network. Nodes 4 and 5 both calculate optimised coverage sets using \( \text{naap()} \). Nodes 4 and 5 determine \( \text{heard}_1 = \{2, 3, 4, 5\} \).

**At Node 4:**

\[
\begin{align*}
O_4 &= N_4 - O_1 - \text{heard}_1 - \{1\} \\
&= \{1, 5, 8, 9, 18\} - \{2, 3, 4, 5\} - \{1\} \\
&= \{8, 9, 18\}
\end{align*}
\]

\[
\begin{align*}
O_4 &= PM(4, O_4, (O_1 \cap N_4)) \\
&= PM(4, O_4, (\{2, 3, 4, 5\} \cap \{1, 5, 8, 9, 18\})) \\
&= PM(4, O_4, \{5\}) \\
&= \{8, 9\}
\end{align*}
\]

The neighbour nodes in \( O_4 \) have equivalent required transmission power. Therefore node 4 will adjust its transmission power to the maximum required transmission power found in \( O_4 \) and not perform the local optimisation.

**At Node 5:**

\[
\begin{align*}
O_5 &= N_5 - O_1 - \text{heard}_1 - \{1\} \\
&= \{1, 4, 6, 9, 14, 15, 18\} - \{2, 3, 4, 5\} - \{1\} \\
&= \{6, 9, 14, 15, 18\}
\end{align*}
\]

\[
\begin{align*}
O_5 &= PM(5, O_5, (O_1 \cap N_5)) \\
&= PM(5, O_5, (\{2, 3, 4, 5\} \cap \{1, 4, 6, 9, 14, 15, 18\})) \\
&= PM(5, O_5, \{4\}) \\
&= \{6, 14, 15, 18\}
\end{align*}
\]
The neighbour nodes in $O_5$ do not have an equivalent required transmission power. Therefore node 5 will use function $localOptimise(O_j)$ to perform a local optimisation. The nodes with minimal required transmission power in $O_5$ are found and their neighbours subtracted from $O_5$ to determine a reduced broadcast set with lower power requirement that introduces an additional hop, thereby ensuring all nodes are reached and reducing the overall power required for transmission from node 5.

$$j = \text{minimumDistanceNode}(T_5)$$
$$= 18$$

$$TempO_5 = PM(5, TempO_5, 18)$$
$$= \{6, 14, 15, 18\}$$

$$TempO_5 = T_5 - \{18\}$$
$$= \{6, 14, 15\}$$

$$O_5 = O_5 + \{18\}$$
$$= \{18\}$$

$$j = \text{minimumDistanceNode}(C_5)$$
$$= 6$$

$$TempO_5 = PM(5, TempO_5, 6)$$
$$= \{6\}$$

$$TempO_5 = TempO_5 - \{6\}$$
$$= \emptyset$$

$$O_5 = O_5 + \{6\}$$
$$= \{6, 18\}$$

Optimisation terminates when $TempO_5$ is an empty set. $O_5$ now contains the locally power optimised minimum set of nodes. The node now adjusts its transmission power
to the maximum power in $O_5$, which is node 6.

The neighbouring nodes of node 5 that are also in the previous optimised set are added to the final optimised set and attached to the packet. This allows nodes receiving a broadcast from node 5 to take account of nodes in node 1's optimised set and the set of neighbours to which they would be broadcasting, which may be located within transmission range.

$$O_5 = O_5 + (N_5 \cap O_1)$$

$$= \{4, 6, 18\}$$

When nodes 6 and 18 receive a broadcast they determine the set $heard_5 = \{6, 18\}$.

At Node 6:

$$O_6 = N_6 - O_5 - heard_5 - \{5\}$$

$$= \{5, 14, 15, 18\} - \{4, 6, 18\} - \{6, 18\} - \{5\}$$

$$= \{14, 15\}$$

$$O_6 = PM(6, O_6, (O_5 \cap N_6))$$

$$= PM(6, O_6, (\{4, 6, 18\} \cap \{5, 14, 15, 18\}))$$

$$= PM(6, O_6, (\{18\}))$$

$$= \{14, 15\}$$

The neighbour nodes in $O_6$ have an equivalent required transmission power. Therefore no local optimisation is performed.
At Node 18:

\[ O_{18} = N_{18} - O_5 - \text{heard}_5 - \{5\} \]
\[ = \{4, 5, 6, 8, 9\} - \{4, 6, 18\} - \{6, 18\} - \{5\} \]
\[ = \{8, 9\} \]

\[ O_{18} = PM(18, O_{18}, (O_5 \cap N_{18})) \]
\[ = PM(18, O_{18}, (\{4, 6, 18\} \cap \{4, 5, 6, 8, 9\})) \]
\[ = PM(18, O_{18}, (\{4, 6\})) \]
\[ = \emptyset \]

Node 18's optimised set is an emptyset, therefore broadcast is inhibited. Similarly nodes 14 and 15 will also have emptysets.

### 3.3.3 NAAP Worked Example Results

Figures 3.2, 3.3 and 3.4 show the results of flooding using Blind flooding, MPR and NAAP. The source of the flood is node 1. The arrows in the figures show the destination nodes for each node's broadcast. In the figures, black directed lines show the effect of broadcasts on neighbouring relays, whereas grey directed lines show the effects of broadcasts on normal nodes. Black coloured nodes are relays, grey coloured nodes are not. In the Blind flooding example it is possible to see the potential for medium contention and packet collisions. Additionally, the problem of duplicate packet reception can also be seen. In MPR this is limited somewhat due to the reduced number of broadcasts. However, when a relay rebroadcasts a packet, the broadcast is also heard by the previous relay as well as neighbouring relays. This is shown by the double directed black lines. When relays are neighbours (nodes 4 and 5), if a random delay of sufficient length is not introduced there is a high possibility of the relays needing to contend for the medium or packet collision occurring. Additionally the neighbouring nodes that they share in common will receive duplicate packets. The use of neighbour awareness and local optimisation combined with power control in NAAP helps to alleviate these problems as seen in Figures 3.1 and 3.4. Table 3.3.3 shows the results of simple analysis of Blind flooding, MPR flooding and NAAP flooding as shown in Figures 3.2, 3.3 and 3.4 respectively. It can be
seen that the use of transmission power control when broadcasting is beneficial as the number of equivalent full power transmission for NAAP is just over half that of MPR and significantly less than Blind flooding. Although NAAP results in 1 more transmission than MPR, the number of received packets is significantly less. A more complete investigation into the performance of NAAP is provided in Section 3.6.

### 3.4 Localised Minimum Spanning Tree Flooding

A proximity graph is a graph, in which vertices are connected by an edge if and only if they satisfy some condition of closeness. In this section, we consider the use of a well known proximity graph called the Minimum Spanning Tree (MST) graph (Toussaint, 1980). The MST graph as shown in Figure 3.5, is a connected graph (path of edges between any two vertices) that uses the minimum total edge length. This results in a graph with one less edge than the number of vertices. The MST has traditionally been used in networks for determining broadcast trees using global topology information.

The Relative Neighbourhood Graph (RNG) is another proximity graph as described in Chapter 2. Its use is described in (Cartigny et al., 2003a) as the basis of optimised flooding algorithm. The MST is a subgraph of RNG and may be computed from the RNG by removing edges that create a cycle in the graph. This results in the formation of a tree or directed acyclic graph from all nodes back to the broadcasting node. Thus the MST generates a more optimal broadcast path than RNG, but suffers as there is no fault tolerance in the resulting graph (Borbash and Jennings, 2002). Fault tolerance refers to the number of alternative paths a message may travel towards a node, thus

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Equivalent Full Power Broadcasts</th>
<th>Transmitted / Received Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blind</td>
<td>18.0</td>
<td>18 / 52</td>
</tr>
<tr>
<td>MPR</td>
<td>9.0</td>
<td>9 / 31</td>
</tr>
<tr>
<td>NAAP</td>
<td>4.9</td>
<td>10 / 21</td>
</tr>
</tbody>
</table>

Table 3.1 Simple Comparison of Blind flooding, MPR and NAAP
In (Li et al., 2003) the authors propose to use the MST algorithm with restricted topology information (one hop) to perform distributed topology control. This is advantageous in ad hoc networks where it is not feasible to have global topology information for the entire ad hoc network.

We have proposed to apply the MST algorithm in a distributed manner to improve the performance of flooding in ad hoc networks. In the distributed MST approach, the topology available to the MST algorithm is restricted to one hop, yet still allows for an optimal broadcast set of nodes with minimal transmission range to be determined as with the centralised approach within that one hop. Importantly, the resulting distributed MST graph does not exhibit the tree like structure of the centralised MST with global topology knowledge. It can be seen by comparing Figures 2.12, 3.5 and 3.6 that MST \( \subseteq \) Localised MST \( \subseteq \) RNG as described in (Li et al., 2003). Thus many of the benefits of MST are maintained with the addition of fault tolerance not found in the centralised approach.

It should be noted that the use of distributed MST for optimised flooding was pro-
posed independently and concurrently by (Cartigny et al., 2003b). Given the low neighbour degree inherent in distributed MST, the authors have extended the use of distributed MST to include directional antennas.

3.4.1 LMSTFlood Algorithm

LMSTFlood works as follows: Each node, upon receiving a broadcast message, calls \textit{LMSTFlood}(message). The algorithm determines if the message has been seen before. If not, then a localised broadcast set (BSET) is determined by supplying the MST algorithm with the broadcasting node’s one hop topology information. Neighbour elimination is used, thus the previous broadcasting node and all neighbouring nodes that may have heard the previous broadcast are removed from the BSET. If the BSET is not an empty set, then the required transmission power to reach all the remaining nodes in the BSET is determined and the message rebroadcast.
Algorithm \textit{LMSTFlood}(message)

1. \textbf{if} not seen message before
2. \hspace{0.5cm} \text{\textit{N1} $\leftarrow$ one hop neighbours}
3. \hspace{0.5cm} \text{\textit{BSET} $\leftarrow$ MST(\textit{j}, \textit{N1})}
4. \hspace{0.5cm} \text{\textit{i} $\leftarrow$ node broadcast was received from}
5. \hspace{0.5cm} \text{\textit{Heard} $\leftarrow$ nodes that received previous broadcast}
6. \hspace{0.5cm} \text{\textit{BSET} $\leftarrow$ \textit{BSET} \setminus \textit{i}}
7. \hspace{0.5cm} \text{\textit{BSET} $\leftarrow$ \textit{BSET} \setminus \textit{Heard}}
8. \hspace{0.5cm} \textbf{if} \textit{BSET} $\neq \emptyset$
9. \hspace{1.5cm} \text{\textit{T}_{\text{power}} $\leftarrow$ maximum\_transmission\_power(\textit{BSET})}
10. \hspace{1.5cm} \text{Broadcast(\textit{Message}, \textit{T}_{\text{power}})}
11. \hspace{0.5cm} \textbf{endif}
12. \hspace{0.5cm} \textbf{endif}
3.4.2 Calculation of MST

The MST algorithm $MST(j, NI)$ called in $LMSTFlood(message)$ is based upon Prim’s algorithm (Prim, 1957). In Prim’s algorithm, the MST tree is ”grown” from a specific node. This is done by repeatedly adding edges of smallest cost and using a priority based queue to store the list of edges that need to be considered. Given a graph $G = (V, E)$, where $V$ is a set of vertices and $E$ a set of edges, then for each vertex $v$ in $V$, we maintain its priority in ”priority[v]” equal to the minimum weight (distance in this case) of any edge $e$ in $E$ connecting $v$ to the partial MST. If there is no edge connecting $v$ to the partial MST then the weight is equal to $\infty$. The parent of each $v$ for an edge of minimum weight is maintained in ”p[v]”. If there is no parent then it is an emptyset. $T$ is the resulting list that contains a set of vertices ({v1, v2}, ...) that form the MST.

Algorithm $MST(j, NI)$

1. $T \leftarrow \emptyset$
2. $Q \leftarrow$ an empty priority queue
3. $V \leftarrow$ all vertices in NI
4. for all $v$ in $V$ do
5. $\text{priority}[v] \leftarrow \infty$
6. \( p[v] = \emptyset \)
7. addPriorityQueue\((Q, v)\)
8. updatePriorityQueue\((Q)\)
9. endfor
10. priority\([j]\) \(\leftarrow\) 0
11. while \( Q \neq \emptyset \) do
12. \( u = \text{getMinimumPriority}(Q) \)
13. if \( p[u] \neq \emptyset \)
14. \( T \leftarrow T \cup \{(p[u], u)\} \)
15. \( A \leftarrow \text{adjacentVertices}(u) \)
16. for each \( v \) in \( A \) do
17. if \( v \) in \( Q \) and \( w(u, v) < \text{priority}[v] \) then
18. \( \text{priority}[v] \leftarrow w(u, v) \)
19. updatePriorityQueue\((Q)\)
20. \( p[v] \leftarrow u \)
21. endif
22. endfor
23. endwhile
24. return \( T \)

The following is an example of finding the MST for the hypothetical node layout in Figure 3.7 using algorithm \( \text{MST}(j, NI) \). Figure 3.7 shows four nodes A, B, C and D along with the distance between them. The edges between the nodes are shown by undirected lines along with their respective distances. We assume node A is the source node from which we intend to find the MST. Algorithm \( \text{MST}(j, NI) \) first calculates the priority queue of each vertex in \( Q \) as (priority, parent).

\[
Q = \{A, B, C, D\} \\
= \{(0, \text{nil}), (\infty, \text{nil}), (\infty, \text{nil}), (\infty, \text{nil})\}
\]

We now remove node A from the priority queue, \( Q \), as it has lowest priority (least distance):
\[ Q = \{B, C, D\} = \{(3, A), (2, A), (3, A)\} \]

We now remove node C from Q as it has the least distance from A. Thus resulting in \( T = \{(A, C)\} \).

\[ Q = \{B, D\} = \{(3, A), (1, C)\} \]

We now remove node D from Q as it has the least distance from A. Thus resulting in \( T = \{(A, C), (C, D)\} \).

\[ Q = \{B\} = \{(3, A)\} \]

We now remove node B from Q as it has the least distance from A. Thus algorithm \( MST(j, NI) \) will return the resulting MST as \( T = \{(A, C), (C, D), (A, B)\} \) and \( Q = \emptyset \). The resulting MST graph is shown in Figure 3.7 by the directed lines.

### 3.4.3 LMSTFlood Example

Figure 3.8 shows the results of distributed MST as calculated by each node and shown in the figure by a thickened black line between the nodes. Node A determines its local MST using algorithm \( MST(j, NI) \) resulting in \( BSET = \{B, C\} \). Node A adjusts its transmission power to include the furthest node in its BSET, in this case node C. A shaded circle represents the adjusted transmission range, whereas a solid lined circle represents the full transmission range. The broadcast from node A is received by nodes B, C and D. Node C has no MST neighbours other than node A that it received the broadcast from. Node C therefore calculates an empty BSET, which inhibits it from rebroadcasting. Node B, receives the broadcast and calculates a BSET = \{D\}. However, node D is eliminated from the BSET of node B as LMSTFlood at node B determines that node D has received the broadcast given the transmission range.
Node D, although not an MST neighbour of node A, receives the broadcast and determines a $BSET=\{B, E\}$. Node B is eliminated as it has received the message resulting is a $BSET=\{E\}$. Node D therefore adjusts its transmission range and broadcasts to node E. Node E is inhibited from rebroadcasting the message as it has no MST neighbours other than node D and therefore calculates an empty BSET.

### 3.5 Simulation Environment

A simulation was developed that generates a random topology of nodes within a 600 meter by 600 meter area. Nodes have a maximum transmission range of 100 meters. Time is divided into epochs. An ideal MAC layer is assumed. There is no medium contention nor hidden-node scenario within the simulation as it is assumed that during an epoch all nodes can complete their transmission. The transmission medium is error free. A bidirectional link between two nodes is assumed upon reception of a beacon message. All neighbouring nodes receive broadcast messages.
if within broadcast range, thus neighbour reachability is always 100 percent. A first order radio model (Heinzelman et al., 2000) as described in Section 3.2 is assumed with a packet length $k$ of 4000 bits.

We have chosen to utilise a simplistic simulation environment without an IEEE 802.11 MAC layer. By doing this we abstract the behaviour and implementation of the flooding mechanism from problems associated with a specific MAC layer. This allows us to more accurately determine the actual behaviour and performance of the optimised flooding mechanisms at limiting the broadcast storm problem. The effects of a more realistic propagation environment, IEEE 802.11 MAC layer and background traffic are examined later in Chapter 5.

3.6 Simulation Results

The aim of the simulation is to compare the performance of NAAP and LMSTFlood with existing flooding mechanisms. The simulation also provides insight into the inherent characteristics of the respective optimised flooding mechanisms.

We select three flooding mechanism for comparison: Blind flooding, MPR, and RNG. Blind flooding is selected as it forms the basis of many reactive and proactive ad hoc routing protocols and does not perform transmission power control. Blind flooding may be seen as a brute force approach to information dissemination as it suffers from worst performance in terms of overhead, but has desirable properties for routing protocols. MPR is selected as it is an optimised flooding mechanism that reduces redundant rebroadcasts, but does not utilise transmission power control and is used in the proactive routing protocol OLSR. An RNG based flooding mechanism is selected as it utilises transmission power control and provides the best performance in terms of overhead compared with existing optimised flooding mechanisms. It must be noted that the use of RNG as the basis of the optimised flooding mechanism RRS (Cartigny et al., 2002a) was proposed around the same time as NAAP.

In each simulation run the simulation determines a random topology. A random node in each topology is selected as the node that initiates the flood. Each random
topology is used to determine the performance of each flooding mechanism. The
topologies generated are not fully connected therefore some topologies may result
in a partitioned ad hoc network. To obtain confidence intervals, the simulation is
executed 100 times starting with a different initial seed. The results are averaged and
95% confidence intervals are generated. The following Figures (3.9 - 3.16) show the
performance of each flooding mechanism as the concentration of nodes is increased.
Each simulation run terminates either upon the reception by all nodes of the packet
being broadcast (complete flood) or when the event queue is empty.

Figure 3.9 shows the total energy in Joules consumed by each optimised flooding
mechanism to perform a complete flood of the ad hoc network. The three mecha­
nisms NAAP, RNG and LMSTFlood consume significantly less energy consumption
than MPR to complete a flood. The reasons for this are as follows:

- From Equation 3.1, we can see that the energy required to transmit a message is
a factor of two variables that are controllable. The first variable is the message
size, k, in bits. The second is distance, $d^\lambda$ where $\lambda = 2$. Thus performing short
range transmissions consume less energy than long range transmissions. As NAAP, RNG and LMSTFlood utilise transmission power control, they benefit from replacing (where possible) full power broadcasts with multiple low power broadcasts.

- Although, the number of packet transmissions may be increased as seen in Figure 3.10, we can see from Equation 3.2 that there is also a significant cost associated with receiving packets. Thus flooding mechanisms should reduce the number of duplicate packets received by nodes during a flood. To reduce this overhead NAAP, RNG and LMSTFlood reduce their transmission power to only include, where possible, those necessary nodes within a broadcast. Therefore there are less duplicate packets experienced by nodes. This is shown in Figures 3.11 and 3.12 where MPR (without transmission power control) has significantly more received and duplicate packets than the flooding mechanisms that utilised transmission power control. As the node density of the network increases, MPR suffers significantly compared to NAAP, RNG and LMSTFlood. Thus the use of transmission power control when broadcasting allows for a reduction in the number packets received by nodes (more importantly the number of duplicate packets which are not useful) as node density increases. It therefore allows for a reduction in power consumption and more effective spatial reuse of the broadcast spectrum.

Figure 3.10 shows the number of transmissions required to complete a flood. All mechanisms show an increase in the number of transmissions with respect to the number of nodes. The rate of growth is lower for MPR than for NAAP, RNG and LMSTFlood. This is partially because NAAP, RNG and LMSTFlood attempt to minimise broadcast distance by introducing additional broadcast hops. NAAP, RNG and LMSTFlood are all able to reduce transmission distance as the node density increases. RNG and LMSTFlood are able to do this more effectively than NAAP as shown in Figure 3.15. LMSTFlood shows less transmissions than RNG, this is a result of there being fewer edges associated with the MST graph compared to the RNG graph (hence less redundancy) as shown in Figures 2.12 and 3.6. Because MPR does not use transmission power control, if the density of nodes increases and
Figure 3.10 A comparison of packets transmitted by NAAP, MPR, LMSTFlood, RNG and Blind flooding.

Figure 3.11 A comparison of packets received by NAAP, MPR, LMSTFlood, RNG and Blind flooding.
Figure 3.12 A comparison of duplicate packets received by NAAP, MPR, LMSTFlood, RNG and Blind flooding.

the network area is maintained then the number of transmissions required to cover all nodes does not grow as quickly as the other mechanisms. This is also evident by the resulting network radius of each mechanism as shown in Figure 3.14.

Figure 3.13 shows the average overhead per broadcast packet in bytes incurred by NAAP and MPR. In MPR the relay set may be distributed through beacon messages and therefore incurs overhead in beacon messages. However, In the simulation, we append the relay set to each packet prior to broadcast as done with source based MPR mechanisms (Peng and Lu, 2002).

RNG and LMSTFlood incur no additional overhead as each mechanism can determine independently whether or not to rebroadcast. It can be seen that as the node concentration increases, the required overhead of NAAP does not grow significantly. The calculation of overhead in the simulations does not include neighbour discovery through beacon messages.

Figure 3.14 shows the resulting network radius in broadcast hops. Routing protocols may benefit from flooding mechanisms that have a lower network radius when per-
Figure 3.13 A comparison of the per packet overhead for NAAP, MPR, LMSTFlood, RNG and Blind flooding.

Figure 3.14 A comparison of resulting network radius in hops for NAAP, MPR, LMSTFlood, RNG and Blind flooding.
forming route discovery as fewer hops introduces less delay. From figure 3.15 we can see that NAAP, RNG and LMSTFlood reduce transmission distance as the node density increases, therefore the network radius will increase for these mechanisms. MPR does not reduce broadcast power nor introduce additional hops and therefore has the lowest network radius. MPR and NAAP may be the most useful to routing protocols as the network radius does not increase dramatically with an increase in node density. Although the network radius of RNG and NAAP is greater than MPR, both mechanisms have the added benefit of reducing the broadcast storm problem more significantly than MPR, thereby providing improved performance. For a proactive routing protocol NAAP and RNG may be more useful given their improved performance, despite the increase in network radius as only link state information is disseminated.

Figure 3.16 shows the node coverage per broadcast with increasing density. As above we see that MPR does not reduce transmission distance and therefore as the node density increases more nodes are covered per broadcast, however as shown in figure 3.12 this results in significant duplicate packet reception. NAAP, RNG and LMSTFlood
Figure 3.16 A comparison of the average node coverage per broadcast for NAAP, MPR, LMSTFlood, RNG and Blind flooding.

are able to restrict broadcast coverage as node density increases. Therefore they tend to be more scalable to higher node densities.

3.7 Adjusting performance of NAAP

As mentioned in Section 3.3, it is possible to adjust the performance of NAAP by setting the RangeLimit constraint in algorithm naap(). RangeLimit specifies a limit to the minimum transmission range below which NAAP will not decompose broadcasts using algorithm localOptimise(Oj). Figures 3.17 - 3.22 show how the performance of NAAP may be varied as the RangeLimit is adjusted from 0% (no minimum distance) to 90% of the maximum transmission range set in the simulations. MPR and LMSTFlood are shown for comparison, with MPR providing an upper bound (worst performance) and LMSTFlood a lower bound (best performance) in terms of reducing the broadcast storm problem.

Figure 3.17 shows the energy consumption of NAAP with varying RangeLimit compared to MPR and LMSTFlood. NAAP shows the least energy consumption when no
Figure 3.17 A comparison of energy consumed by NAAP for varying RangeLimits.

Figure 3.18 A comparison of packets transmitted by NAAP for varying RangeLimits.
Figure 3.19 A comparison of duplicate packets received for NAAP with varying RangeLimits.

Figure 3.20 A comparison of average network radius in hops for NAAP with varying RangeLimits.
**Figure 3.21** A comparison of average transmission distance for NAAP with varying RangeLimits.

**Figure 3.22** A comparison of average node coverage per broadcast for NAAP with varying RangeLimits.
limit is placed upon the minimum distance for decomposition of broadcasts. NAAP shows comparable energy consumption to MPR when $RangeLimit$ is set to 90%. This shows that NAAP benefits greatly from the use of algorithm $localOptimise(O_j)$, which allows for a single full power transmission (by a single relay node) to be replaced by two or more successive low power transmissions (over additional relays). Minimal difference in performance is achieved when the $RangeLimit$ is varied between 0% and 30%. As NAAP relies upon neighbour elimination based mechanisms, it must include additional information within the broadcast packet to ensure that it is able to reduce transmission power and still limit redundant broadcasts. Introducing many low power broadcasts results in NAAP performing additional transmissions. Some of these transmissions may be redundant transmissions as neighbour elimination provides only limited benefit given that NAAP information attached to broadcast packets is based upon the preceding two hops. Thus given reduced transmission ranges, it is possible for broadcasting nodes to include some nodes within a broadcast that need not be included.

Figure 3.18 shows that the number of transmissions performed decreases as $RangeLimit$ is increased. This is to be expected as broadcasts of increased transmission range (figure 3.21) are performed resulting in a smaller network radius (figure 3.20) and more node coverage per broadcast (figure 3.22). As $RangeLimit$ increases the number of duplicate packets received also increases as shown in figure 3.19. Thus the ability of NAAP to reduce the broadcast storm problem decreases and approaches that of MPR.

The use of $RangeLimit$ makes it possible to increase the $RangeLimit$ in NAAP and thus accommodate situations where a lower network radius is required, such as with routing protocols or sparsely populate ad hoc networks. Thus NAAP may be used in a variety of applications and its performance adjusted accordingly.

### 3.8 Conclusions

In ad hoc networks the process of disseminating information throughout the network forms the basis of routing protocols, network management, service discovery and
information collection. As packet transmission in ad hoc networks is broadcast in nature, it is therefore important that information dissemination be done with minimal effect to the network. In this chapter we have introduced two new optimised distributed flooding mechanisms: Neighbour Aware Adaptive Power (NAAP) flooding and Localised Minimum Spanning Tree Flooding (LMSTFlood). Both mechanisms utilise transmission power control to reduce the broadcast storm problem.

NAAP is a distributed flooding mechanism that utilises local two hop neighbour knowledge obtained through the exchange of beacon messages. NAAP employs several mechanisms (neighbour coverage, power control, neighbour awareness and local optimisation) to limit the broadcast storm problem. The performance of NAAP may be adjusted by limiting the minimum transmission range over which NAAP decomposes broadcasts.

LMSTFlood is a distributed flooding mechanism that calculates the Minimum Spanning Tree (MST) from local one hop neighbour knowledge as opposed to NAAP and Multipoint Relay (MPR) flooding, which require two hop neighbour knowledge. This allows LSMTFlood to select an optimal broadcast set of nodes with minimal transmission range. The resulting distributed MST graph does not exhibit the tree like structure of the centralised MST graph with global topology knowledge. Thus many of the performance benefits of centralised MST are maintained in the distributed MST with the addition of fault tolerance not found in the centralised MST approach. The use of MST allows each node to be self selecting and thus determine whether or not it is required to rebroadcast a packet or not.

NAAP and LMSTFlood are compared through simulation with Blind flooding, MPR and a Relative Neighbourhood Graph (RNG) based flooding mechanisms. The performance of the mechanisms is based upon: energy consumption, transmitted packets, received duplicate packets, packet overhead, resulting network radius, transmission range and node coverage per broadcast.

In terms of reducing the broadcast storm problem optimised flooding mechanisms that utilise transmission power control (NAAP, LMSTFlood and RNG) are shown to provide a significant performance improvement over optimised flooding mecha-
nisms, such as MPR, that do not utilise transmission power control.

NAAP is shown to have performance that ranges between RNG and MPR. LMSTFlood is shown to have the best performance of the optimised flooding mechanisms in terms of reducing the broadcast storm problem. However, MPR has the lowest network radius of the optimised flooding mechanisms - roughly equivalent to that of Blind flooding. MPR is shown to not scale well to high node densities as it does not benefit from transmission power control. However, MPR’s low network radius may be beneficial in routing protocols where less delay may be required. But, this comes at the price of higher overhead in terms of the broadcast storm problem when compared to NAAP, LMSTFlood and RNG. MPR consumes more energy per flood than the other optimised flooding mechanisms that benefit from transmission power control.

NAAP is not quite as efficient as RNG or LMSTFlood, but benefits as its performance in terms of network radius and coverage may be adjusted by adjusting the minimum transmission range below which broadcasts are decomposed. NAAP may therefore be used in varying situations from routing protocols to general information dissemination.

LMSTFlood significantly reduces energy consumption, utilises a smaller average transmission range and results in nodes receiving less duplicate packets during a flood. It is thus more effective at limiting the broadcast storm problem than existing optimised flooding mechanisms. But due to the resulting high network radius and the per hop delays introduced in routing, LMSTFlood is not particularly suited to reactive routing protocols. However, it may be useful in proactive routing protocols for disseminating link state information or for general information dissemination - where the resulting routes and network radius may have less affect.
Chapter 4

Resource Aware Information Dissemination

4.1 Introduction

The very nature of ad hoc networks ensures that devices are mobile in nature, hence they must be small and light enough for users to carry. These restrictions require that the devices have portable yet limited energy supplies. Exceptions are physical infrastructure or mobile devices that are attached to a reliable energy source. Given the multitude of network enabled devices currently available, it is safe to say that ad hoc networks will be composed of heterogeneous devices. Thus devices may have varying characteristics and constraints.

This chapter introduces two new flooding mechanism: Utility Based Multipoint Relay Flooding (UMPR) and Utility Based Flooding (UBF). Both mechanisms are distributed optimised flooding mechanism for ad hoc networks. However, both mechanisms (unlike existing optimised flooding mechanisms) account for the available resources of devices within an ad hoc network while optimising the flooding process so as to address the broadcast storm problem. UMPR extends Multipoint Relay (MPR) flooding to allow for a limited degree of resource awareness. This is achieved by selecting those relay nodes, without unique neighbours, based upon their available resources. UMPR shows improved performance compared to existing optimised flooding mechanisms. However, through simulation UMPR was found to have de-
ficiencies as the majority of relay nodes have unique neighbours and are therefore selected irrespective of their available resources. UBF was developed to address these deficiencies by making the selection of all relays based only upon a node's available resources.

Resource Aware Information Dissemination is the process of disseminating information in such a way that mechanisms are aware of and utilise available resources within the network in an efficient and aware manner. In Chapter 2 the broadcast storm problem is introduced and mechanisms for optimised flooding are described that reduce the broadcast storm problem. However, they do not address the need for resource awareness. These two requirements may possibly oppose each other as an optimal flood is not necessarily the most resource aware flood.

Besides physical constraints such as power supplies, communication devices may also have user based constraints imposed upon their behaviour and participation in an ad hoc network. A user with a laptop, may be willing to participate in information dissemination as long as the laptop is attached to a permanent power source or the remaining battery power is sufficient. However, in situations where no alternative power source is available other than a battery with limited remaining power, a user may place constraints inhibiting the participation of their laptop in the ad hoc network. Equally it makes sense for mechanisms to strive to extend the operational lifetime of devices within the network. This can be achieved through optimisation, but also by utilising devices that are most suitable based upon their resources and constraints.

In this chapter we compare the performance of UMPR and UBF to Blind flooding and Multipoint Relay flooding (MPR). We show that the introduction of a utility based mechanism that accounts for available node resources and constraints is beneficial. The performance of the proposed mechanisms in terms of flood reachability over successive floods is significantly better than an optimised flooding mechanism such as MPR, while still achieving comparable performance in terms of reducing the broadcast storm problem.
4.2 Utility Based Multipoint Relay Flooding

In this section we propose an extension to MPR called Utility Based Multipoint Relay (UMPR) that introduces limited selection of relays based upon a forwarding utility ($U_f$). The forwarding utility is representative of a node’s remaining battery power and hence its usefulness as a relay.

In MPR flooding (Qayyum et al., 2001), a broadcasting node must select a set of relays that are responsible for rebroadcasting a message thereby ensuring a flood propagates. These relays are selected from the broadcasting node’s list of one hop neighbours. The first relays selected by a broadcasting node are those nodes that have one or more neighbouring nodes (two hop nodes of the broadcasting node), that are not reachable from any other relay. These two hop nodes are termed “unique neighbours”. Thus any one hop nodes with unique neighbours are immediately selected as relays irrespective of their available resources. The remaining relays in MPR are selected based upon their total number of neighbours or some other resource blind criteria.

UMPR allows for the selection of the remaining relays from those neighbouring nodes (without unique neighbours) to be a function of a node’s remaining battery power, thereby allowing for a limited degree of resource awareness in the selection of relays.

4.2.1 UMPR - Forwarding Utility

UMPR extends MPR by allowing nodes that do not have unique neighbours to be selected as relays based upon a forwarding utility, $U_f$ (Equation 4.1). The forwarding utility is a function of a node’s remaining battery power utility, $U_p$ (Equation 4.2), and a neighbour utility, $U_n$ (Equation 4.3). Thus an element of resource awareness is introduced into the optimised flood. This is important as node’s selected as relays must have sufficient battery power to rebroadcast messages.

$$U_f = U_p U_n$$  (4.1)
The utility $U_p$ (Equation 4.2) specifies the remaining internal battery power of a device. A sigmoid function as seen in Figure 4.1 is used to determine the utility as it provides a good estimate of the required behaviour (low utility and slow change at low power; sharp change in utility at medium power; high utility and slow change at high power). $P_i$ is the remaining internal battery power of a device and is mapped onto the sigmoid. To shift the sigmoid function accordingly, $s$ is defined as half the value of the full power of a node’s battery.

The neighbour utility $U_n$ (Equation 4.3) for a node $i$ is equal to the number of unallocated nodes in the two hop pool that are neighbours of node $i$ divided by the total number of neighbours of node $i$. Therefore node $i$’s utility will decrease as its shared neighbours are allocated to other relays. The neighbour utility favours those nodes with more un-allocated neighbours thus encouraging the selection of relays.

Figure 4.1 Sigmoid Power Utility $U_p$

$$U_p(i) = \frac{1}{1 + e^{-P_i+s}}$$  \hspace{1cm} (4.2)

$$U_n(i) = \frac{\text{unallocated two hop neighbours of node } i}{\text{total two hop neighbours of node } i}$$  \hspace{1cm} (4.3)
that share fewer neighbours. By comparison MPR only considers the total distinct two hop neighbours.

4.2.2 The UMPR Algorithm

In this section we describe the implementation of UMPR. We assume that each node maintains a list of its one hop neighbours and two hop neighbours. This information is acquired through the exchange of beacon messages. Additional information such as remaining battery levels may also be exchanged through beacon messages. Upon receiving a broadcast message, it is possible to determine the previous broadcasting node \((i)\) and the previous relay set \((R_i)\) from the message header. Note, in algorithm \(UMPR()\), we denote the neighbours of a node \(i\) as \(N(i)\).

A node \(j\) using UMPR generates a pool of one hop \((P_1)\) and two hop \((P_2)\) neighbouring nodes. All nodes that are neighbours of the previous broadcasting node \(i\) are removed from both pools. Nodes in \(P_1\) with unique neighbours in \(P_2\) are removed from \(P_1\) and added to the set of relays \((R_j)\). Thus all one hop neighbours of the relays in \(R_j\) are removed from \(P_2\). The remaining nodes in \(P_1\) with no unique neighbours are assigned a forwarding utility \((U_f)\). An allocation then occurs, which adds the node in \(P_1\) with the highest utility to the set of relays and removes its neighbours from \(P_2\). The forwarding utility for the remaining nodes is then revised. This continues until \(P_1\) or \(P_2\) is an empty set. The set of chosen relays is then attached to the broadcast message. Nodes not in the attached relay set are inhibited from rebroadcasting.

We describe algorithm \(UMPR()\) as follows with its associated line numbers in the algorithm:

1. Upon receiving a broadcast, determine all one hop and two hop neighbours that did not receive the previous broadcast. \((\text{lines 1-6})\)

2. Add those one hop nodes with unique neighbours as relays to \(R_j\) and remove them from the pool of one hop neighbours. Remove the one hop neighbours of the new relays from the pool of two hop neighbours. \((\text{lines 7-9})\)

3. Calculate a forwarding utility \(U_f\) for each remaining one hop neighbour. Add
the one hop neighbour with the highest utility to the relay list and remove any two hop neighbours that will receive its broadcast. Remove the new relay from the list of one hop neighbours. *(lines 10-19)*

4. Repeat from step 3 until all two hop neighbours are covered.

### Algorithm UMPR()

1. $P_1 \leftarrow$ 1-hop neighbours
2. $P_2 \leftarrow$ 2-hop neighbours
3. $i \leftarrow$ last node to broadcast
4. $R_i \leftarrow$ relay set attached to broadcast packet
5. $P_1 \leftarrow P_1 - N(i) - \{i\}$
6. $P_2 \leftarrow P_2 - N(i) - N(R_i) - \{i\}$
7. $R_j \leftarrow$ nodes in $P_1$ with unique neighbours in $P_2$
8. $P_1 \leftarrow P_1 - R_j$
9. $P_2 \leftarrow P_2 - N(R_j)$
10. **while** $P_1 \neq \emptyset$ or $P_2 \neq \emptyset$
    11. **for** each node in $P_1$
    12. Calculate its forwarding utility
    13. **endfor**
    14. $n \leftarrow$ highest utility node from $P_1$
    15. $P_2 \leftarrow P_2 - N(n)$
    16. $R_j \leftarrow R_j + n$
    17. $P_1 \leftarrow P_1 - n$
    18. Remove nodes from $P_1$ with $U_f = 0$
19. **endwhile**
20. **return** $R_j$

### 4.2.3 Example and Explanation of UMPR flooding

Figure 4.2 shows an example of UMPR flooding with the following neighbour arrangement: $N(A)\{B,C,D\}$, $N(B)\{A,H,I\}$, $N(C)\{A,F,G,H,I\}$, $N(D)\{A,E,F,G\}$.

The broadcasting node $A$ calculates a pool of one hop neighbours $P_1\{B,C,D\}$ and
Node E is a distinct neighbour of node D.

(a) UMPR – Node E is a unique neighbour of node D.

Node D is selected as a relay first due to the presence of the distinct neighbour node E. This is irrespective of the forwarding utility of node D.

(b) UMPR – Node D selected as a relay. Calculate Uf for nodes B and C.

Node B is selected as a relay based upon its forwarding utility, which is higher than that of node C.

(c) UMPR – Node B selected as relay. Node C inhibited from broadcasting.

Figure 4.2 Utility Based Multipoint Relay Flooding Example
two hop neighbours $P_2$={E,F,G,H,I}. Node D has a unique neighbour (node E) that is not reachable from any other possible relays (nodes B or C) as shown in Figure 4.2(a). Node D is therefore added to the set of relays, $R_A$={D}. Neighbouring nodes of node D are removed from $P_2$ resulting in $P_2$={H,I} and $P_1$={B,C} as in MPR.

As no more node's have unique neighbours, node A then calculates forwarding utilities for the remaining nodes in $P_1$ as shown in Figure 4.2(b).

**Node B’s forwarding utility is calculated as follows:**

$$P_B = 6$$

$$s = \frac{\text{Battery Power Range}}{2} = \frac{10}{2} = 5$$

$$U_p(B) = \frac{1}{1 + e^{-P_B + s}} = 0.73$$

$$U_n(B) = \frac{2}{2} = 1$$

$$U_f(B) = U_p(B) \times U_n(B) = 0.73$$
Node C's forwarding utility is calculated as follows:

\[ P_C = 3 \]

\[ s = \frac{Battery\ Power\ Range}{2} = \frac{10}{2} = 5 \]

\[ U_p(C) = \frac{1}{1+e^{-pC+s}} = 0.12 \]

\[ U_n(C) = \frac{2}{4} = 0.5 \]

\[ U_f(C) = U_p(C) \times U_n(C) = 0.12 \times 0.5 = 0.06 \]

Node B has a higher forwarding utility than node C, therefore the remaining nodes in \( P_2 \) are allocated to node B resulting in \( R_A = \{B,D\} \) as shown in Figure 4.2(c).

If MPR had been used, node A would have always selected node C and node D or node B and node D as forwarding nodes. This selection would have occurred despite node C's low remaining battery power. Multiple floods from node A would have depleted node C or node B. UMPR allows for both nodes to be used alternatively as their remaining battery power decreases.

From the simulation results in section 4.5 it can be seen that UMPR improves upon the flood reachability provided by MPR flooding as shown in Figure 4.6. UMPR reduces the broadcast storm problem, however its performance in terms of flood reachability only approaches that of Blind flooding. Blind flooding is able to achieve its flood reachability as it relies upon a brute force approach (all nodes propagate) to dissemination, but suffers from the broadcast storm problem. In Figure 4.7, we can see that Blind flooding causes more nodes to enter a restricted state (as described in section 4.4) more quickly than MPR or UMPR. Additionally, there are only slightly more restricted nodes in UMPR than MPR. From this we can deduce that UMPR is unable to fully utilise available resources compared to a brute force approach such as
Blind flooding.

The reason UMPR is unable to fully utilise resources is as follows: UMPR relay nodes are selected using a forwarding utility. However, this utility based selection is only applied to those nodes that do not have unique one hop neighbours. One hop nodes with unique neighbours are selected as relays irrespective of their available resources. Their neighbouring nodes are removed from the set of two hop neighbours prior to a forwarding utility being calculated for the remaining possible relays (without unique neighbours). In some cases when using UMPR, the process of selecting suitable relays from the remaining one hop neighbours based upon their forwarding utility may not occur. This is because the majority of relays selected may all have unique neighbours and are therefore selected without consideration of their resources. Figure 4.3 is obtained from the simulations described in section 4.4. Figure 4.3 shows both the total number of relays selected during a complete flood and the total number of those relays that were selected because they had a unique neighbour. It can be seen that the number of relays with unique neighbours constitutes a significant majority of the total relays selected. Thus selecting relays from the remaining nodes based upon their resources provides only limited gain.
In the next section, we propose a new resource aware flooding mechanism that does not rely upon the selection of relays because they have unique neighbours as is done in MPR and UMPR. Instead, nodes are selected as relays based solely upon their forwarding utility.

4.3 Utility Based Flooding

In this section we introduce Utility Based Flooding (UBF). UBF extends UMPR so as to make the relay selection process fully resource aware. This is achieved in three ways: (i) UBF selects relays based solely upon their forwarding utility. (ii) The selection of nodes with unique neighbours is not a priority. These nodes are allowed to broadcast, but are only selected after those nodes with the best forwarding utility. In this way the coverage of the best nodes is accounted for prior to those nodes with unique neighbours as is done in MPR and UMPR. (iii) We represent any constraints imposed upon a device by its benevolence.

The forwarding utility as used in UBF may be a function of various resource based and constraint based utilities. The UBF forwarding utility, $U_f$ (Equation 4.4), used to select relays is resource aware in that it is a function of various utilities that model the usefulness of nodes as relays.

$$U_f = B U_p U_n$$

(4.4)

The utility $U_p$ (Equation 4.2) represents the utility of a node based upon its remaining internal battery power as described for UMPR. In UBF we have defined the benevolence utility ($B$). Benevolence represents the user based constraints that may be imposed upon a node. A user may allow a device attached to a reliable power source to fully participate in network activities. However, if the device is mobile and the battery power drops below a specified limit, the user may not wish the device to participate. Existing flooding mechanisms do not account for this type of behaviour. Thus their performance will be degraded in such a network as they may select restricted nodes as relays which will not rebroadcast messages. Other utilities which
account for link or node stability, device load or other user defined parameters may also be used. A node may be either active, restricted or depleted. Thus a node's benevolence depends upon which state it is in. These states are described in more detail in Section 4.4.

In UMPR the neighbour utility, $U_{n}$, is used as a mechanism to limit the selection of relays that are neighbours by choosing those relays that share as few one hop neighbours in $P_1$ as possible. However, in UBF the neighbour utility plays an additional role. UBF relies upon the neighbour utility to increase the utility of those possible relays that may have unique neighbours. However, if a node with a unique neighbour is not suitable because of constraints or low resources, then its utility will remain low such that it may only be selected after all other possible relays are selected. Thus UBF will maximise the coverage of two hop nodes based upon the forwarding utility. This is beneficial as UBF is able to ensure that where possible the best relays are selected to ensure continuation of the optimised flood.

The node's forwarding utility may be zero in the following situations:

- The constraints (user) imposed upon a device, result in its benevolence (B) being zero.
- The number of unallocated neighbours of a node is zero, thus its neighbour utility will be zero. In essence the node provides no additional coverage.

Given the finite power supply of mobile devices, mechanisms such as MPR (and UMPR to a lesser extent) will only utilise nodes to continue a flood that provide the best broadcast coverage and therefore will more quickly deplete these nodes. This is especially problematic in slow or reasonably static networks where these nodes may be selected repeatedly. The use of a forwarding utility for selecting all relays allows UBF to distribute the load of optimised flooding among those nodes most suited based upon their resources. Over multiple broadcasts, UBF is thus able to adapt to changes in node resources and constraints, where existing optimised flooding mechanisms would fail.
4.3.1 The UBF Algorithm

In this section we describe the implementation of the UBF algorithm. We assume that each node maintains a list of its one hop neighbours and two hop neighbours. This information is acquired through the exchange of beacon messages. Additional information such as the remaining battery levels may also be exchanged through beacon messages. Upon receiving a broadcast message, it is possible to determine the previous broadcasting node \( i \) and the previous relay set \( R_i \) from the message header. In the algorithm \( UBF() \) we denote the neighbours of a node \( i \) as \( N(i) \).

A node using UBF generates a pool of one hop neighbours \( (P_1) \) and a pool of two hop neighbours \( (P_2) \). All nodes that may have received the previous broadcast based upon two hop neighbour knowledge are removed from \( P_1 \) and \( P_2 \). One hop neighbouring nodes with neighbours in \( P_2 \) are assigned a forwarding utility. An allocation then occurs, which adds the node in \( P_1 \) with the highest forwarding utility to the relay set \( (R_j) \) and removes its one hop neighbours from \( P_2 \). The forwarding utility for the remaining nodes in \( P_1 \) is then revised. Nodes in \( P_1 \) with a zero forwarding utility are removed from \( P_1 \). The selection of relays and the re-evaluation of forwarding utilities for possible relays continues until either \( P_1 \) or \( P_2 \) is an empty set. The final relay set is attached to the broadcast message. Nodes not in the relay set are inhibited from rebroadcasting.

The UBF algorithm may be stated as:

1. Upon receiving a broadcast, determine all one hop \( (P_1) \) and two hop \( (P_2) \) neighbours that did not receive the previous broadcast. (lines 1-6)

2. Calculate a forwarding utility \( U_f \) for each node in \( P_1 \), then select the node with the highest forwarding utility. Remove any nodes in \( P_2 \) that are neighbours of the new relay node. Add the new relay to the relay set \( (R_j) \) and remove it from \( P_1 \). Remove nodes in \( P_1 \) with zero forwarding utility. (lines 7-16)

3. Repeat from step 2 until either \( P_1 \) or \( P_2 \) is an empty set.
Algorithm $UBF()$

1. $P_1 \leftarrow$ 1-hop neighbours
2. $P_2 \leftarrow$ 2-hop neighbours
3. $i \leftarrow$ last node to broadcast
4. $R_i \leftarrow$ relay set attached to broadcast packet
5. $P_1 \leftarrow P_1 - N(i) - \{i\}$
6. $P_2 \leftarrow P_2 - N(i) - N(R_i) - \{i\}$
7. while $P_1 \neq \emptyset$ or $P_2 \neq \emptyset$
   8. for each node in $P_1$
      9. Calculate its forwarding utility
   10. endfor
   11. $n \leftarrow$ highest utility node from $P_1$
   12. $P_2 \leftarrow P_2 - N(n)$
   13. $R_j \leftarrow R_j + n$
   14. $P_1 \leftarrow P_1 - n$
   15. Remove nodes from $P_1$ with $U_f = 0$
16. endwhile
17. return $R_j$

4.3.2 Example and Explanation of UBF flooding

In this section, we provide examples of UBF in Figures 4.4-4.5 and compare the selection of relays in UBF to UMPR and MPR.

In Figure 4.4 and Figure 4.5, two examples of UBF flooding are shown. Node $A$ initiates the flood and must select a set of its one hop neighbour nodes as relays to continue the flood. The figures have the following neighbour arrangement: $N(A) = \{B,C,D\}$, $N(B) = \{A,H,I\}$, $N(C) = \{A,F,G,H,I\}$, $N(D) = \{A,E,F,G\}$. Node $A$ calculates a pool of one hop $P_1 = \{B,C,D\}$ and two hop neighbours $P_2 = \{E,F,G,H,I\}$. Node $A$ then calculates forwarding utilities for each node in $P_1$ as as shown in Figure 4.4(a). To simplify the following example, we utilise a benevolence ($B = 1$) in Equation 4.4 and therefore the forwarding utility is $U_f = U_p U_n$ in the following examples. The following calculations are based upon Figure 4.4:
(a) UBF – Calculating Forwarding Utilities for one hop nodes

Node B has the highest forwarding utility and is selected as a relay by node A.

(b) UBF – Node B selected as relay. Recalculate Forwarding Utilities of one hop nodes.

Node D has a higher forwarding utility than node C and is therefore selected as a relay. This is different from MPR/UMPR, where node D would be selected first due to the presence of a distinct node E.

(c) UBF – Nodes B and D selected as relays. Node C inhibited from rebroadcasting

Figure 4.4 Utility Based Flooding Example 1
Node B has the highest forwarding utility and is selected as a relay by node A.

Node C has a higher forwarding utility than node D and is chosen as a relay prior to node D, despite node D having a unique neighbour. UBF selects relays based solely upon their forwarding utility.

Node D has lowest forwarding utility and a unique neighbouring node E.

Node D is allowed to broadcast to cover its unique neighbour. However, given the low battery power, there is a possibility node D will not rebroadcast. This may occur when node A has outdated neighbour information. Thus those nodes with greatest coverage and battery power are selected first.

Figure 4.5 Utility Based Flooding Example 2
Node B’s forwarding utility is calculated as follows:

\[ P_B = 6 \]

\[ s = \frac{BatteryPowerRange}{2} = \frac{10}{2} = 5 \]

\[ U_p(B) = \frac{1}{1 + e^{-P_B + s}} = 0.73 \]

\[ U_n(B) = \frac{2}{2} = 1 \]

\[ U_f(B) = U_p(B) * U_n(B) * B = 0.73 \]

Node C’s forwarding utility is calculated as follows:

\[ P_C = 3 \]

\[ s = \frac{BatteryPowerRange}{2} = \frac{10}{2} = 5 \]

\[ U_p(C) = \frac{1}{1 + e^{-P_C + s}} = 0.12 \]

\[ U_n(C) = \frac{4}{4} = 1 \]

\[ U_f(C) = U_p(C) * U_n(C) = 0.12 * 1 = 0.12 \]

Node D’s forwarding utility is calculated as follows:

\[ P_D = 5 \]

\[ s = \frac{BatteryPowerRange}{2} = \frac{10}{2} = 5 \]
\[
U_p(D) = \frac{1}{1 + e^{-r_D^{++}}}
\]
\[= 0.5\]

\[
U_n(D) = \frac{2}{3} = 1
\]

\[
U_f(D) = U_p(D) \times U_n(D)
\]
\[= 0.5\]

Nodes in \(P_2\) are allocated to the node in \(P_1\) with the highest utility. Node \(B\) has the highest forwarding utility, therefore nodes \(I\) and \(H\) are allocated to node \(B\) and removed from \(P_2\) as shown in Figure 4.4(b). Node \(B\) is added to the relay list resulting in \(R_A = \{B\}\).

Node \(A\) then recalculates the forwarding utilities given the previous allocation as shown in Figure 4.4(b).

**Node C’s forwarding utility is calculated as follows:**

\[P_C = 3\]

\[s = \frac{\text{BatteryPowerRange}}{2}
\]
\[= \frac{10}{2} = 5\]

\[
U_p(C) = \frac{1}{1 + e^{-r_C^{++}}}
\]
\[= 0.12\]

\[
U_n(C) = \frac{2}{4} = 0.5
\]

\[
U_f(C) = U_p(C) \times U_n(C)
\]
\[= 0.12 \times 0.5 = 0.06\]
Node D’s forwarding utility is calculated as follows:

\[ P_D = 5 \]

\[ s = \frac{BatteryPowerRange}{2} = \frac{10}{2} = 5 \]

\[ U_p(D) = \frac{1}{1+e^{-PD+s}} = 0.5 \]

\[ U_n(D) = \frac{3}{3} = 1 \]

\[ U_f(D) = U_p(D) * U_n(D) = 0.5 \]

In Figure 4.4, node D is selected as its overall forwarding utility is greater than node C’s. Nodes E, F and G are allocated to node D and removed from \( P_2 \) resulting in an empty set, as shown in Figure 4.4(c). The final relay set for node A is \( R_A = \{B, D\} \).

If MPR had been used, node A would have allocated node D as a relay first, due to the existence of the unique neighbour node E. Additionally either node B or node C would have been selected as a relay depending upon the implementation of MPR (usually based upon neighbour coverage and node identifier), despite node C’s low internal battery power. Thus multiple floods from node A would then have depleted node C. If UMPR had been used, node A would have selected node D as a relay due to the unique neighbour, and then selected either node B or C as a relay depending on their forwarding utility.

In Figure 4.4(c), it is important to note that if node D’s battery power was depleted or lower than node C’s, then given the UBF algorithm, node C would be selected as a relay prior to node D. This scenario is shown in Figure 4.5. In UBF, relays are selected based solely upon their forwarding utility. Relays with the highest forwarding utility are thus selected (with more priority) before those with a lower forwarding utility. This can be seen in Figure 4.5(a), where node B has the greatest forwarding utility and is selected as a relay first. Its one hop neighbours are removed from the
pool of two hop nodes, resulting in $P_2=\{E,F,G\}$. The forwarding utilities are recalculated and node C is then selected as a relay prior to node D (as its forwarding utility is still greater than that of node D). Node C's one hop neighbours (nodes $G$ and $F$) are removed from the pool of two hop nodes, resulting in $P_2=\{E\}$. This is different from the first example in Figure 4.4 where node D is selected prior to node C. $P_2$ contains only node $E$, thus when forwarding utilities are recalculated, all potential relays (except node D) will have a zero neighbour utility ($U_n$).

The major difference between UBF and UMPR, is that potential relays (such as node D) are not selected based upon the presence of a unique neighbour (node E). Nodes with unique neighbours, and low forwarding utility are the last to be selected as relays. This is because the emphasis in UBF is on choosing those nodes as relays that are most suited to rebroadcasting based upon their forwarding utility (which in this case is a function of a node’s remaining battery power). Allowing relays to be selected which may have only unique neighbour nodes ensures coverage of those neighbouring nodes. However, this is not a priority. In the case where node D fails to rebroadcast, node C would still rebroadcast the message to nodes $F$ and $G$. This ensures the flood propagates. In the case where node D does rebroadcast the message, then there is additional overhead produced. However, this increases the ability of UBF to reach more nodes in the ad hoc network despite the presence of restricted nodes and inaccurate neighbour information.

### 4.4 Simulation Environment

In this chapter we developed a simulation environment that allows for nodes to have different levels of battery power. Nodes are initialised with randomly varying battery power between 0 joules and 2 joules of energy. Each node is also constrained by varying user based constraints which are dependent upon a node’s remaining battery power. A node may be in one of either three states:

- **Active** - the node participates in all operations of the ad hoc network and will rebroadcast and receive packets accordingly.
Resource Aware Information Dissemination

- **Restricted** - the user has specified a minimal battery level below which the node will not participate in packet flooding, but may still use the network and receive broadcast packets.

- **Depleted** - the node has ceased operation due to battery failure and therefore cannot participate in the network.

The simulation generates a random topology of nodes within a 600 meter by 600 meter area. Nodes have a maximum transmission range of 100 meters. Time is divided into epochs. An ideal MAC layer is assumed. There is no medium contention nor hidden-node scenario within the simulation as it is assumed that during an epoch all nodes can complete their transmission. The transmission medium is error free. A bidirectional link between two nodes is assumed upon reception of a beacon message.

A first order radio model (Heinzelman et al., 2000) is assumed. In this model the first order radio dissipates $E_{elec} = 50nJ/bit$ to run the circuitry of a transmitter or receiver and a further $e_{amp} = 100pJ/(bit*m^2)$ for the transmitter amplifier. Equation 4.5 is used to calculate the costs of transmitting a $k$-bit message a distance $d$. Equation 4.6 is used to calculate the costs of receiving a $k$-bit message.

$$E_{Tx}(k, d) = E_{elec} * k + e_{amp} * k * d^2$$ (4.5)

$$E_{Rx}(k) = E_{elec} * k$$ (4.6)

### 4.5 Simulation Results

The aim of the simulations is to compare the ability and performance of Blind flooding, MPR, UMPR and UBF at adapting to changes in the network. As each flood progresses through the ad hoc network it consumes energy resources at each node due to transmission and reception of packets. This may either result in nodes becoming restricted or depleted. Thus these nodes will not participate in rebroadcasting.
As nodes in the active state may have varying energy resources, it is beneficial for the flooding mechanism to adapt and utilise those nodes with the greatest resources (where possible), thereby extending the life of the network.

Blind flooding is neither optimised nor resource aware, it is a brute force approach to disseminating information irrespective of nodes’ available energy resources. MPR is optimised and relies upon selected relay nodes to rebroadcast. The problem with the MPR approach is that these relays are not selected in a resource aware manner. UMPR and UBF are both optimised flooding mechanisms. UMPR is only partially resource aware in its selection of relays, while UBF is fully resource aware in its selection of relays.

In each simulation run the simulation determines a topology, initiates 100 successive floods and waits for each flood to complete or no more simulation events. After each flood, the number of nodes reached by the flood is recorded. A random node in the topology is selected as the initial node of each flood. To obtain confidence intervals, the simulation is executed 100 times starting with a different initial seed so to remove any correlation between simulation results.
Figure 4.6 shows the \textit{reachability} achieved (percentage of nodes reached) by each flooding mechanism over 100 successive floods. All mechanisms achieve 100 percent reachability for the first flood. However, reachability over successive floods should decrease as nodes' battery power resources decrease and they enter ”restricted” or ”depleted” states. The results show that UBF is able to maintain above 90 percent reachability up to 70 successive floods. Blind flooding, which is a brute force approach, is able to maintain above 90 percent reachability up to 42 successive floods. UMPR is able to maintain above 90 percent reachability up to 39 successive floods, while MPR only achieves 23 successive floods.

These results suggest that UBF and UMPR are able to adapt to changes in node state and resources, thereby utilising only those nodes suited to continuing a flood as shown in Figures 4.2, 4.4 and 4.5. UMPR is not as successful as UBF, however UMPR provides better reachability than MPR. UMPR also shows comparable performance to Blind flooding, up to 39 successive broadcasts. Blind flooding achieves its performance due to its brute force approach, but suffers due to significant overhead. This can be seen as UMPR’s performance at delivering packets degrades more gradually than Blind flooding. This is because Blind flooding results in more nodes being depleted per flood. The use of a forwarding utility in UBF and UMPR allows these mechanisms to account for a node’s remaining battery power when determining relays. UMPR however is limited in it application of the forwarding utility due to its selection of relays that have unique neighbours, thereby limiting its performance. MPR shows the worst performance as it an optimised approach that selects only those nodes with the best coverage, irrespective of their available resources or constraints.

The ability to direct the responsibility of flooding to nodes most suited allows for the load of flooding to be shared by all nodes, thus increasing the use of all nodes rather than just those optimal nodes (relays), which may be running low on battery power or be constrained by a user. Blind flooding’s brute force approach results in all capable nodes participating in a flood, thus increasing the possibility of a flood progressing even in the presence of restricted nodes. However, in Figure 4.6, Blind flooding shows a rapid decrease in broadcast reachability between 40 and 60 broadcasts. This decrease in reachability is a direct result of the increased number of restricted nodes
(75%) as shown in Figure 4.7. MPR and UMPR, however, do not show a significant increase in restricted nodes. This is because the number of nodes receiving broadcasts is lower than that of UBF or Blind flooding. In the case of UMPR, it shows that UMPR is unable to fully utilise all nodes. This is because UMPR allocates one hop neighbour nodes with unique neighbours as relays and does not determine their suitability to continuing a flood. Because UBF attempts to distribute the load of flooding, the number of restricted nodes is higher than MPR, but lower than Blind flooding with its brute force approach. At the same time, UBF is able to maintain a much higher broadcast reachability than either UMPR, MPR or Blind flooding due to its ability to select only those nodes most suitable to continuing the flood.

Figures 4.8, 4.9, 4.10 and 4.11 show the performance (energy consumption, transmitted packets, duplicate packets and packet overhead) of the flooding mechanisms. The figures show the results obtained for a single broadcast over increasing node densities. The comparison with MPR is to show that there is a minimal increase in overhead given utility based forwarding decisions. From the figures it can be seen that that UMPR, UBF and MPR have equivalent performance providing significant reductions for a completed broadcast in terms of power consumption, packets
transmitted and duplicate packets received over Blind flooding. However, UBF and UMPR have significantly better performance than MPR in terms of delivering packets to more nodes in an resource constrained ad hoc network.

4.6 Conclusions

Ad hoc networks are composed of heterogeneous devices that are mobile in nature, hence the devices must be small and light enough for users to carry. Thus devices are restricted by portable yet limited energy supplies unless they are attached to an external source of energy. Given the heterogeneous nature of devices, they may have varying characteristics and constraints.

Resource Aware Information Dissemination is the process of disseminating information in such a way that mechanisms are aware of and utilise available resources within the network in an efficient and aware manner. In literature there exist mechanisms for reducing the broadcast storm problem. However, the need for resource awareness is not addressed. These two requirements may possibly oppose each other as the op-
Figure 4.9 A comparison of packets transmitted per completed flood vs increasing number of nodes.

Figure 4.10 A comparison of duplicate packets received per completed flood vs increasing number of nodes.
timal flood in terms of reducing the broadcast storm problem, is not necessarily the most resource aware flood.

In this chapter we have introduced Utility Based Multipoint Relay (UMPR) flooding and Utility Based Flooding (UBF). Both are distributed optimised flooding mechanisms for ad hoc networks that unlike existing optimised flooding mechanisms account for the available resources of devices within ad hoc networks while optimising the flooding process so as to address the broadcast storm problem.

UMPR extends Multipoint Relay (MPR) flooding by introducing a degree of resource awareness in the selection of relays. UMPR performs the same initial step as MPR, by selecting neighbour nodes as relays if they have a unique neighbour that is two hops from the broadcasting node. However, UMPR differs from MPR in that the remaining relays without unique neighbours are selected based upon their forwarding utility. In MPR these relays are selected depending on the coverage they provide. Thus UMPR compared MPR is able to show improved performance in terms of flood reachability and comparable performance in terms reducing the broadcast storm problem.

![Figure 4.11 A comparison of the additional per packet overhead vs increasing number of nodes.](image)
UBF extends UMPR by removing the first step as implemented in MPR. This is important as simulation results show that the number of nodes selected as relays based solely upon the existence of unique neighbours accounts for a significant portion of the total relays selected during a flood. Thus UBF extends UMPR by requiring that the selection of all relays be based solely upon each node's forwarding utility. This allows for UBF compared to UMPR, MPR and Blind flooding to show significantly improved performance in terms of flood reachability and comparable performance in terms reducing the broadcast storm problem. UBF further extends UMPR by accounting for any user based constraints that may be imposed upon a device thereby limiting its benevolence within the ad hoc network.
Chapter 5

More Reliable Information Dissemination

5.1 Introduction

Mechanisms for information dissemination in ad hoc networks, such as Blind flooding, form an integral part of communication in ad hoc networks. Blind flooding is seen as a reliable flooding mechanism as all nodes participate in rebroadcasting a message, thereby ensuring its propagation throughout an ad hoc network. However, Blind flooding results in the broadcast storm problem. Numerous optimised flooding mechanisms have been proposed to reduce problems associated with the broadcast storm problem. However, reducing the broadcast storm problem reduces the inherent redundancy found in Blind flooding, thus making the optimised flooding mechanisms less reliable.

In this chapter we compare the performance of optimised flooding mechanisms and Blind flooding at reliably delivering a message in the presence of increasing background traffic. We show that Blind flooding is remarkably robust and is able to reliably deliver messages in the presence of increasing background traffic. However, Blind flooding suffers from the broadcast storm problem. Optimised flooding mechanisms that are aimed at reducing the broadcast storm problem prove to be less reliable in the presence of increasing background traffic than Blind flooding.
Optimised flooding mechanisms rely upon selected nodes to rebroadcast messages during a flood thereby ensuring that the flood propagates. But, given the use of unreliable broadcast transmissions in optimised flooding mechanisms, a problem arises when nodes responsible for rebroadcasting a message do not actually receive the message.

There exists a need for optimised flooding mechanisms that are able to reduce the broadcast storm problem and also improve the reliability. Existing reliable flooding mechanisms provide only limited optimisation and require significant overhead to ensure reliability. Some of these mechanisms require that acknowledgements are returned to the source of a flood, however this is not always necessary depending upon the application. Particularly in a typical ad hoc network where the source of a flood may not know of the existence of non local nodes.

In this chapter we introduce Reliable Minimum Spanning Tree (RMST) flooding. RMST is a reliable and optimised flooding mechanism that takes advantage of the unique nature of the Minimum Spanning Tree (MST) as used by LMSTFlood in Chapter 3. RMST utilises unicast transmission (with link layer acknowledgement and retransmission), which provides more reliable packet delivery than broadcast transmission as used by existing optimised flooding mechanisms. Reliability is improved at each transmitting node, thus RMST distributes the load of ensuring flood reliability among all nodes. We compare the reliability and performance (in terms of reducing the broadcast storm problem) for RMST, LMSTFlood, MPR and Blind flooding in the presence of increasing background traffic. Our results show that RMST provides comparable performance in terms of reducing the broadcast storm problem when compared to existing optimised flooding mechanisms. Importantly, RMST shows comparable reliability, in terms of packet delivery, to Blind flooding and significantly improves upon the level of reliability provided by existing optimised flooding mechanisms.
5.2 Reliable Minimum Spanning Tree (RMST) Flooding

Reliable Minimum Spanning Tree (RMST) flooding is a reliable, distributed and optimised flooding mechanism for ad hoc networks. RMST utilises the Minimum Spanning Tree (MST) algorithm with local one hop neighbour knowledge in a distributed manner as is done in LMSTFlood (Chapter 3). The MST is used by each node to determine those closest neighbouring nodes that it must include within any transmissions, to ensure a connected graph, thereby ensuring a flood propagates throughout an ad hoc network.

RMST extends LMSTFlood by utilising the unique nature of the distributed MST that results in a connected graph with minimum neighbour degree of 1, maximum neighbour degree of 6 and an average neighbour degree of less than 2.04 nodes (Li et al., 2003). Given that the prior broadcasting node is included in this average and may therefore be removed, the average resulting Broadcast Set (BSET) for the MST is thus reduced to 1.04 nodes. This low neighbour degree can be seen in the distributed MST as shown in Figure 5.1.

The use of the distributed MST algorithm in LMSTFlood allows for a highly opti-
mised flood, thus greatly reducing the broadcast storm problem. However, there ex­
ists a significant problem in broadcast environments where a broadcast transmission
may be lost due to packet corruption, packet collision or hidden node transmissions.
Thus the use of broadcast transmission is not reliable. Therefore it is possible that
nodes may not receive a broadcast transmission. Furthermore those nodes that do not
receive a broadcast transmission may be required to receive a transmission in order
for the flood to propagate. This is especially true in the case of optimised flooding
mechanisms, where selected nodes are responsible for retransmission. Given that op­
timised flooding mechanisms greatly reduce the redundancy found in Blind flooding,
there may be situations where a packet may be lost and a flood may not propagate
because of the low degree of redundancy.

The use of distributed MST and the resulting small BSET allows for unreliable IEEE
802.11 broadcast transmissions (as used by existing flooding mechanisms) to be re­
placed with more reliable IEEE 802.11 unicast transmissions. Unicast transmission
is more reliable than broadcast transmission as unicast transmission implements a
RTS/CTS exchange at the MAC layer prior to transmission in order to reduce prob­
lems associated with the hidden node problem. More importantly, unicast trans­
mission utilises a frame retransmission mechanism at the MAC layer based upon a
positive acknowledgement scheme (ARQ) as described in Chapter 2. Thus, a trans­
mitting node will retransmit a frame if it does not receive a positive acknowledge­
ment from the destination node. The IEEE 802.11 ARQ is not completely reliable
and packet loss is still possible. However it provides a more reliable transport mecha­
nism than broadcasting and requires no modifications to the MAC layer. The number
of retransmissions before a timeout occurs may be adjusted, but is generally 4-7 re­
transmissions. If a node fails to retransmit a message to a destination node, it is able
to detect the failure and may utilise an alternative scheme (as described in Section
5.2.3) to continue dissemination.

RMST is aimed at making the process of optimised flooding in ad hoc networks
less susceptible to packet loss and therefore more reliable. This is achieved in RMST
through the use of more reliable unicast transmission (as opposed to unreliable broad­
cast transmission) combined with the distributed MST algorithm, thereby increasing
reliability and reducing the broadcast storm problem.

5.2.1 RMST Algorithm

The algorithm \texttt{RMST}(message) is called by each node upon receiving a unicast message at a specific port address. The \texttt{RMST}(message) algorithm first determines whether or not the message has been received before. If the message has not been received before then the current node must determine its local MST. Algorithm \texttt{MST}(j, NI) (described in Section 3.4.2) is called to determine a BSET by supplying it with the current node’s one hop neighbour knowledge at line 2 in \texttt{RMST}(message).

At line 3 in \texttt{RMST}(message), the previous node from which the unicast message was received, is removed from the BSET. The BSET now contains the set of MST neighbours that the current node is responsible for unicasting the message to. If the BSET is not an empty set, then RMST determines the required transmission power of each node in the BSET and unicasts the message to the destination node with the required transmission power (lines 5-7 in \texttt{RMST}(message)). It is possible to select the order in which the transmissions occur. In our simulations we elected to unicast to the furthest node first and the closest node last. The reason being that nodes further away may move out of transmission range in a mobile environment. If a unicast fails, then \texttt{RMST}(message) enters a recovery stage as described in Section 5.2.3, in order to ensure the flood propagates.

Algorithm \texttt{RMST}(message)

1. \textbf{if} not seen message before
2. \hspace{1cm} \texttt{BSET} ← \texttt{MST}(1-hop Neighbours)
3. \hspace{1cm} i ← last node message was unicasted from
4. \hspace{1cm} \texttt{BSET} ← \texttt{BSET} − i
5. \hspace{1cm} \textbf{for} each node \texttt{j} in \texttt{BSET}
6. \hspace{2cm} \texttt{T}_{\text{power}} ← \texttt{transmission\_power}(j)
7. \hspace{2cm} \texttt{ack} ← \texttt{Unicast(Message, T}_{\text{power})}
8. \hspace{2cm} \textbf{if} NOT \texttt{ack}
9. \hspace{2cm} \texttt{recovery(Message)}
Algorithm \(MST(j, NI)\)

1. \(T \leftarrow \emptyset\)
2. \(Q \leftarrow \) an empty priority queue
3. \(V \leftarrow \) all vertices in \(N1\)
4. \textbf{for} all \(v\) in \(V\) \textbf{do}
5. \hspace{1em} \(\text{priority}[v] \leftarrow \infty\)
6. \hspace{1em} \(p[v] = \emptyset\)
7. \hspace{1em} \(\text{addPriorityQueue}(Q, v)\)
8. \hspace{1em} \(\text{updatePriorityQueue}(Q)\)
9. \textbf{endfor}
10. \(\text{priority}[j] \leftarrow 0\)
11. \textbf{while} \(Q \neq \emptyset\) \textbf{do}
12. \hspace{1em} \(u = \text{getMinimumPriority}(Q)\)
13. \hspace{1em} \textbf{if} \(p[u] \neq \emptyset\)
14. \hspace{2em} \(T \leftarrow T \cup \{(p[u], u)\}\)
15. \hspace{1em} \(A \leftarrow \text{adjacentVertices}(u)\)
16. \hspace{1em} \textbf{for} each \(v\) in \(A\) \textbf{do}
17. \hspace{2em} \textbf{if} \(v\) in \(Q\) and \(w(u, v) < \text{priority}[v]\) \textbf{then}
18. \hspace{3em} \(\text{priority}[v] \leftarrow w(u, v)\)
19. \hspace{3em} \(\text{updatePriorityQueue}(Q)\)
20. \hspace{3em} \(p[v] \leftarrow u\)
21. \hspace{2em} \textbf{endif}
22. \hspace{1em} \textbf{endfor}
23. \textbf{endwhile}
24. \textbf{return} \(T\)
5.2.2 RMST Example

In Figure 5.2 the distributed MST graph for a topology of nodes is shown as calculated from node A. Node A’s 1-hop topology is obtained through beacon messages and includes nodes B, C, D, E and F. In the LMSTFlood approach, node A would adjust its broadcast transmission power to include the distance of the furthest node, in this case node D. However in RMST node A must first unicast the message being flooded to node D and then unicast the message to node B with the required transmission power to reach each node. As is described earlier, in our simulations we chose to unicast to the furthest node first and the closest node last. Each unicast is shown by a black directed line. If a unicast is successfully received (determined by the reception of an acknowledgement at the link layer) then RMST will unicast the message to the next furthest node, in this case node B.

In Figure 5.2, both unicast messages are successfully delivered. However, when node B unicasts to node F and node D unicasts to node E, both MAC frames are lost or corrupted at the MAC layer as no acknowledgement is returned to the unicasting nodes. Therefore, each node’s MAC layer given the IEEE 802.11 specification will retransmit the frames as shown by the dashed black directed lines until an ACK (as shown by the dashed grey directed line) is received from the destination node or the maximum number of retransmissions at the MAC layer are reached. In the next section, we describe a recovery mechanism to ensure the flood propagates if the MAC layer fails to transmit a frame successfully to a destination node.

5.2.3 RMST Recovery from Failed Unicast Transmissions

In RMST a transmitting node must unicast the message being disseminated to each node within its broadcast set. However, under certain circumstances a node may either have moved away, have been switched off or simply not be able to receive the transmitted message due to interference. Thus, in these situations, the transmitting node is unable to propagate the flood in the intended direction. To recover from such situations RMST may enter a recovery stage whereby it attempts to propagate the flood in the required direction. As shown in Figure 5.3, recovery in RMST is achieved by the following steps:
Figure 5.2 RMST flood utilising IEEE 802.11 unicast and link layer ARQ

Figure 5.3 RMST Recovery
More Reliable Information Dissemination

1. **Unicast to 2-hop MST Neighbours** - Node A fails to unicast a message to its MST neighbour (node D). It therefore determines the MST neighbours of node D (two hop MST Neighbours), if they are one hop neighbours of node A, and unicasts the message to each of these nodes.

2. **Broadcast to all Neighbours** - If Node A fails to reach the two hop MST neighbours as in step 1, then node A may perform a full power broadcast transmission. Receiving nodes determine if their MST neighbours have received the broadcast. If not, then the receiving node must continue the RMST flood by unicasting to its one hop MST neighbours.

5.3 **Simulation Environment**

To determine the performance of RMST, we utilise two different simulations.

The first simulation is performed using the GloMoSim 2.03 (Bajaj et al., 1999) simulation package. GloMoSim provides a complete TCP/IP stack with a Constant Bit Rate (CBR) traffic generator, the AODV routing protocol and an implementation of the IEEE 802.11 MAC with more realistic physical layer. We have modified the GloMoSim 802.11 MAC layer to allow transmission power control for broadcast and unicast packet transmission as required my LMSTFlood and RMST. The results of this simulation are shown in Figure 5.4. It shows the performance of RMST compared to existing flooding mechanisms in the presence of background traffic generated by three CBR source-destination pairs and the AODV routing protocol.

The second simulation is performed using the same simulation as used in previous chapters. The simulation assumes an ideal NULL MAC layer. There is no medium contention nor hidden-node scenario within the simulation as it is assumed that all nodes are able to successfully complete their transmissions. The transmission medium is error free. A bidirectional link between two nodes is assumed upon reception of a beacon message. A first order radio model (Heinzelman et al., 2000) is used to calculate power consumption. In this model the radio dissipates $E_{elec} = 50nJ/bit$ to run the circuitry of a transmitter or receiver and a further $E_{amp} = 100pJ/(bit*m^2)$
for the transmitter amplifier. Equation 5.1 is used to calculate the costs of transmitting a $k$-bit message a distance $d$. Equation 5.2 is used to calculate the costs of receiving a $k$-bit message. The radios have power control and consume the minimal required energy to reach the intended recipients. We consider RMST’s power consumption, packets received, packets transmitted and duplicate packet reception. The results of this simulation are shown in Figures 5.5 - 5.9.

$$E_{Tz}(k, d) = E_{elec} * k + E_{amp} * k * d^2$$  

(5.1)

$$E_{Rx}(k) = E_{elec} * k$$  

(5.2)

The reasoning for the use of a simplified simulation environment is to determine the performance of RMST in terms of reducing the broadcast storm problem. It allows the abstraction of the issues associated with an IEEE 802.11 MAC layer and the GloMoSim signal propagation environment which may result in packet loss and frame retransmissions. Additionally, implementation of an optimised flooding mechanism requires the exchange of neighbour information through beacon messages. This has an added effect upon the performance given the periodic exchange of beacons, however we only wish to concentrate on the actual performance of the flooding mechanisms. The use of a simplified simulation environment allows for more focus upon actual packet transmission and reception performance of RMST without the underlying problems associated with realistic implementations.

The following are the environment settings as used by both simulations. The simulation area is 600 meters by 600 meters. Nodes are placed in a random topology within this area. Nodes have a maximum transmission range of 100 meters. A node within each random topology is selected randomly as the source of a flood. The topologies generated are not fully connected thus some topologies may result in a partitioned ad hoc network. The total number of nodes reachable for each topology is determined so as to account for partitioning.
5.4 Simulation Results

To obtain simulation results in each simulation environment, simulations are run 100 times with a different seed for each run. The final results are averages and 95% confidence intervals are displayed in each graph. We selected Blind flooding, MPR (source based) and LMSTFlood as comparison flooding mechanisms. Blind flooding was selected as it is a brute force approach to flooding that provides a high degree of reliability, but suffers from the broadcast storm problem. MPR and LMSTflood were selected as they are both optimised flooding mechanisms that reduce the broadcast storm problem in ad hoc networks. LMSTFlood implements transmission power control when broadcasting to limit the number of nodes affected by a broadcast whereas MPR does not utilise transmission power control. In Chapter 3, we show that LMSTFlood provides a significant performance in terms of reducing the broadcast storm problem compared to MPR. It is therefore expected that given the reduction in redundancy of the optimised flooding mechanisms that their ability to successfully disseminate information in the presence of background traffic will be affected.
Figure 5.4 shows the percentage of nodes that receive a message being flooded as the Constant Bit Rate (CBR) packet rate is increased. In a NULL MAC environment, delivery is assumed to be 100%. However, in GlomoSim the use of a more realistic IEEE 802.11 MAC and transmission medium, results in packets being lost due to collision, corruption and fading. We utilise three CBR source-destination pairs in the simulation to create background traffic that may effect the delivery performance of the flooding mechanisms. The source-destination pairs are selected randomly and UDP packets of 512 bytes are transmitted between nodes using the AODV routing protocol. Each source begins transmitting data at a random time prior to the initiation of a flood by the flooding mechanism being tested. It is important to note that a flood only last for a short period of time. This period of time is much less than that assigned to each CBR traffic generator.

It is possible to utilise alternative traffic sources (ie. "on/off" or "bursty" traffic sources). However, the use of a CBR traffic sources ensures that the network is loaded and allows for the load of the network to be increased in a controlled fashion. As an average optimised flood of an ad hoc network may last less than 100ms and we are only using a limited number of traffic sources, it is unlikely that the use of "on/off" traffic generator would have any affect. Hence, CBR traffic is sufficient for our experimentation.

Importantly, the use of multiple Blind floods creates a situation in the network where only broadcast transmission (as described in Chapter 2) is used. This is not realistic. The use of CBR traffic and AODV as the routing protocol results in a combination of broadcast and unicast transmission, which is more realistic.

From Figure 5.4, it can be seen that Blind flooding and RMST provide the best delivery performance and are only slightly affected by background traffic. Blind flooding provides reliability through redundant broadcasts, but suffers from the broadcast storm problem as shown in Figures 5.5 - 5.9. RMST, however is an optimised flooding mechanism and therefore limits the broadcast storm problem. RMST is able to achieve comparable delivery to Blind flooding as it utilises more unicast packet transmission compared to broadcast packet transmission which is less reliable. Other optimised flooding mechanisms LMSTFlood and MPR suffer in delivery as broadcast
packets collide with the background traffic. The reasons being that both LMSTFlood and MPR rely upon specific nodes receiving a broadcast. In the case of LMSTFlood, nodes are able to determine whether they are required to rebroadcast by calculating their local MST. But, if a node does not receive a broadcast message then the flood is effectively halted in that direction. Source based MPR mechanisms, as used in the simulation, attach a relay list to the broadcast message. If a message is not received by one or more selected relays, then no rebroadcast will occur effectively cancelling the propagation of the flood at that point.

The results obtained in Figure 5.4 are verified by results obtained in (Cartigny et al., 2003a) through simulation. The authors show experiments with multiple simultaneous optimised floods occurring for RNG and MPR. The results show (for a 512 byte packet being flooded) that as the number of simultaneous floods occur, the reachability of both MPR and RNG optimised flooding mechanisms is severely affected. Importantly, and in line with results obtained in Figure 5.4, RNG is more affected than MPR. From Chapter 3, it can be seen that RNG is more optimised than MPR but less optimised than LMSTFlood. Hence RNG's reachability performance should still be less than MPR but greater than LMSTFlood. Thus Figure 5.4 is accurate as it shows LMSTFlood having worse reachability performance than MPR.

Figure 5.5 shows the power consumed by each mechanism to complete a flood. RMST utilises more energy to complete a flood than LMSTflood. This is to be expected as RMST must perform more transmissions (Figure 5.6) than LMSTFlood, thus resulting in more duplicate and received packets (Figure 5.7). Compared to Blind flooding and MPR, RMST shows significantly better performance in terms of reducing the broadcast storm problem. In Figures 5.7 and 5.8 we see that MPR receives significantly more packets than RMST. The use of transmission power control in RMST when unicasting allows for a reduction in the number packets received by nodes (including duplicate packets which are not useful), despite RMST performing more transmissions. As with LMSTFlood, the use of transmission power control is beneficial as it limits the number of nodes that will hear a transmission. Thus allowing for a reduction in power consumption and more effective spatial reuse of the shared wireless medium.
Figure 5.5 A comparison of the energy consumed by nodes for a complete flood with increasing number of nodes.

Figure 5.6 A comparison of the packets transmitted by nodes for a complete flood with increasing number of nodes.
Figure 5.7 A comparison of the packets received by nodes for a complete flood with increasing number of nodes.

Figure 5.8 A comparison of the duplicate packets received by nodes for a complete flood with increasing number of nodes.
Figure 5.6 shows the number of transmissions required by each mechanism to complete a flood. All mechanisms show an increase in the number of transmissions with respect to the number of nodes. MPR performs less transmissions than LMSTFlood and RMST. This is a result of LMSTFlood and RMST utilising more low power transmissions thereby introducing additional hops. As MPR does not use transmission power control, if the density of nodes increases but the network area is maintained then the number of transmissions required to cover all nodes does not grow as quickly as RMST. This can be seen by the resulting network radius as shown in Figure 5.9.

Thus RMST, which extends LMSTFlood by replacing unreliable broadcast transmission with more reliable unicast transmission, is able to significantly improve the reliability of optimised flooding. RMST is able to achieve comparable performance in terms of reliability to Blind flooding without suffering from the broadcast storm problem. Importantly, RMST significantly outperforms MPR in terms of reducing the broadcast storm problem and has comparable performance to LMSTFlood.
5.5 Conclusions

Mechanisms for information dissemination in ad hoc networks, such as Blind flooding, are a vital part of ad hoc network communication mechanisms. However, Blind flooding in a wireless broadcast environment (an ad hoc network) is subject to the broadcast storm problem. To limit the broadcast storm problem, numerous optimised flooding mechanisms have been proposed in ad hoc network literature. Optimised flooding mechanisms limit the broadcast storm problem by reducing the inherent level of redundancy found in Blind flooding. Thus the unreliable nature of broadcast packet transmission as used by optimised flooding mechanisms results in these mechanisms being more susceptible to failure. We show through experimentation, that although optimised flooding mechanisms reduce problems associated with the broadcast storm problem, they experience significantly bad reachability performance (packet delivery) when increasing levels of background CBR traffic are introduced into the ad hoc network.

Various mechanisms for reliable flooding have been proposed in literature. However, they either suffer from significant messaging overhead in order to disseminate information and also determine whether or not a flooded message was received, or they require modifications to the IEEE 802.11 MAC layer to improve broadcast delivery between nodes.

We have introduced Reliable Minimum Spanning Tree (RMST) flooding. RMST is a distributed yet more reliable optimised flooding mechanism that takes advantage of the unique nature of the distributed Minimum Spanning Tree (MST) and requires no modification to the IEEE 802.11 MAC layer. If given the local topology of a node it is possible using the MST algorithm to determine the closest minimal set of nodes to which a broadcasting node (during a flood) must rebroadcast to ensure a flood propagates. The average size of this forwarding set is shown experimentally to be 2.04 nodes. This allows unreliable broadcast transmission to be replaced by more reliable unicast transmission in RMST, without greatly affecting overhead performance in terms of the broadcast storm problem. Reliability is increased because IEEE 802.11 unicast transmission incorporates RTS/CTS, positive acknowledgement and packet
retransmission. Thus the use of unicast transmission in a distributed flooding mechanism, such as RMST, allows for an increase of flooding reliability in a distributed manner.

We have shown that RMST compared to LMSTFlood, MPR and Blind Flooding is able to reliably deliver packets given a scenario with three source-destination pairs generating CBR traffic. In fact the performance of the optimised flooding mechanisms were shown to suffer in the presence of CBR traffic as the packet rate was increased. However, RMST was able to provide equivalent performance to existing optimised flooding mechanisms in terms of reducing the broadcast storm problem, while ensuring comparable packet delivery performance to Blind flooding without suffering from the broadcast storm problem.
Chapter 6

Resource Aware Information Collection

6.1 Introduction

In ad hoc networks there is a need for all-to-one protocols which allow for information collection or "sensing" of the state of the network and the nodes that form the network. In office or conference network scenarios, the ability to collect information from nodes forming the ad hoc network may be used for service discovery, autconfiguration, network management, topology discovery or data retrieval. Information collection could be used in wartime to collect information from soldiers, such as: bio-metric data, location information, neighbouring soldier information and other local information such as temperature. In disaster recovery, information collection may be of significant importance. In these situations rescuers may wish to ascertain the damage to a network by collecting information from all surviving nodes within the network. From a human perspective, rescuers may be able to detect the presence of survivors by collecting information (possibly bio-metric data) from devices attached to humans. In ad hoc network literature there has been a strong focus on information dissemination protocols for routing (one-to-one) and flooding (one-to-all), but little on information collection for sensing (all-to-one) mechanisms.

There is a parallel between this type of sensing in ad hoc networks and that of sensor networks (Akyildiz et al., 2002). However, ad hoc networks and sensor networks
differ in their application, construction, characteristics and constraints. The number of nodes in a sensor network may be significantly higher in number and have greater density, be more prone to failure, have frequent topology change and more severe device constraints such as limited processing, limited power and low bandwidth communication. The main priority of sensor networks is for the flow of information from sensors back to the sink. However in an ad hoc network sensing may be of secondary importance. Sensing in an ad hoc network during setup and collection of data must be performed in such a way as to have little effect on the normal operation of the ad hoc network. For these reasons, protocols suitable to ad hoc networks are not necessarily suitable to sensor networks and vice versa.

In this chapter we propose Resource Aware Information Collection (RAIC) an information collection mechanism that benefits from the unique use of an optimised resource aware flooding mechanism. RAIC aims to reduce the overhead associated with information collection in ad hoc networks by taking advantage of neighbour knowledge available to nodes. Information collection mechanisms rely strongly upon information dissemination as means of disseminating requests for information and also the formation of reverse paths (backbones) for the flow of information back towards the source of the information request. The process of information collection may be composed of two phases: setup and capture. The setup phase in a sensor network occurs from a state of no knowledge and therefore requires the use of a Blind flooding mechanism to disseminate requests and create paths back to the sink (self organisation). However, in an ad hoc network we can assume there is knowledge of neighbouring nodes acquired from a routing protocol or through the exchange of beacon messages. This allows for information collection mechanisms in ad hoc networks to utilise optimised and resource aware flooding mechanisms as described in Chapter 4.

6.2 Resource Aware Information Collection

In this section we describe Resource Aware Information Collection (RAIC), a distributed, resource aware, two phase (setup and capture) approach to information col-
lection in ad hoc networks. RAIC allows for efficient and optimised information collection from multiple nodes in an ad hoc network to a single node acting as the sink.

Many information collection mechanisms in sensor networks rely upon information dissemination as means of disseminating requests for information. However, sensor networks differ from ad hoc networks in that nodes are deployed randomly and may start from a state of zero knowledge of their immediate environment. Thus sensor network mechanisms must rely upon Blind flooding as a basic mechanism for disseminating requests for information and self organisation. However, in ad hoc networks we may assume there is knowledge of neighbouring nodes acquired from a routing protocol or through the exchange of beacon messages. This allows for information collection mechanisms in ad hoc networks to utilise optimised flooding mechanisms thereby reducing both the overhead associated with flooding and the broadcast storm problem.

RAIC benefits from available local neighbour knowledge through the use of Utility Based Flooding (UBF) as described in Chapter 4. Thus RAIC differs from existing sensor network mechanisms as the use of UBF during the setup phase allows for RAIC to:

- disseminate requests for information in an optimised and resource aware manner.
- build a directed acyclic backbone of relay nodes for relaying collected information back to the sink during the capture phase.
- select relay nodes based upon their utility such that they are suited to relaying collected information back to the sink.
- determine a reverse path utility thus allowing for information to flow back to the sink via the best available relays.
6.2.1 Setup Phase

Figure 6.1 shows the setup and capture phases in RAIC. A sink (node 1) initiates a flood using UBF to disseminate a sensory request (SREQ) message throughout the ad hoc network. SREQ messages are shown as light dashed directed lines. Black nodes are those nodes selected as relays whereas grey nodes are not relays. The length of the lines are representative of the distance between the nodes.

In the RAIC setup phase, UBF is used to disseminate SREQ messages throughout the ad hoc network in an optimised and resource aware manner. The use of an optimised flooding mechanism reduces problems associated with the broadcast storm problem and reduces power consumption due to transmitted and received packets. Relay nodes are selected based upon their forwarding utility, $U_f$ (Equation 4.4). The forwarding utility determines a node's ability to act as a relay in both setup and capture phases. Thus only those nodes most suited, based upon their available resources (such as battery power), are selected. This is important as node's selected as relays
must be capable of continuing a flood and also supporting the flow of information back to the sink.

Resource awareness in terms of a node's internal battery power is expressed as the utility function $U_p$ (Equation 4.2, described in Section 4.2). Additionally, user-based constraints of node participation are represented and described as benevolence ($B$). Other utilities which account for link or node stability, device load or other user-defined parameters may also be used. The neighbour utility $U_n$ (Equation 4.3, described in Section 4.3) allows for a node $i$'s utility to decrease as its shared neighbours are allocated to other relays.

Each node maintains a list of parent nodes and their sink-degree (the hop distance from the sink). The sink-degree is attached to SREQ messages. As SREQ messages are propagated throughout the ad hoc network by relays, nodes may receive more than one SREQ message and therefore have more than one parent. In order to aid in the discovery of possible parent relays, nodes may delay replying to or forwarding SREQ messages. This delay allows for a relay when calculating the reverse path utility to determine any neighbouring relays. Additionally it allows for a child node to discover other, potentially more ideal, parent relays. Alternatively, as the relay list is attached to the SREQ message, it is possible for a child node to avoid this delay and just rely upon the relay list. However, this approach will only return those relays of the preceding parent relay. The formation of the parent-child structure combined with the sink-degree creates a sensory backbone (directed acyclic graph) with a reverse gradient path from all child nodes back to their parents and ultimately back to the sink. The final formation of such a backbone is shown in Figure 6.1 as a solid line between nodes "1" and "2", and between nodes "1" and "3".

During the setup phase, relays calculate a reverse path utility $U_{rp}$ (Equation 6.1) which is attached to the SREQ message to be rebroadcast. $U_{rp}$ specifies the utility of a relay to forward a SREP message back towards the sink. A node's $U_{rp}$ is a function of its own internal remaining battery power (Equation 4.2), total number of parent relays ($|PR|$) of a lower sink degree and the sum of the reverse path utilities of all parent relays (PR).
The reverse path utility is important as it allows for relays during the capture phase to select the best path (if more than one exists) back to the sink. This has similarities with the mechanism used in SARS (Sohrabi et al., 2000), where nodes are elected as relays to form a path back to the sink based upon their additive QoS. Sensors may then select a path with the best QoS and energy reserves when sending data back to the sink. However, RAIC allows nodes to perform this operation on a hop by hop basis as the SREP message is routed back to the sink. The reverse path utility can be seen in Figure 6.2 where the path (nodes: f, d, c, a) from node f back to the sink (node a) is shown by a thicker directed line.

6.2.2 Capture Phase

Figure 6.1 shows RAIC's capture phase. The capture phase occurs when nodes begin replying to received SREQ messages by sending a unicast sensory reply (SREP) messages to their parent relay. The solid directed lines show the flow of SREP messages. As the number of child nodes may be significantly more than parent nodes, child nodes chose the parent node to which they are closest when unicasting a SREP message. In Figure 6.1, node 6 and node 4 both receive multiple SREQ messages. Node 6 selects node 2 as its parent relay and node 4 selects node 3 as its parent relay. Nodes 5, 7 and 8 only receive one SREQ message and therefore are not able to select...
an alternative parent. This process of child nodes selecting their closest parent is important when combined with transmission power control. It allows for a reduction in a node’s power consumption due to packet transmission and isolates the broadcast effects on neighbouring nodes when a node unicasts its SREP message to its selected parent relay.

To reduce the number of packets flowing back to the sink, relays may implement various mechanisms, based on merge and time semantics (directives from the sink attached to the SREQ specifying the type of information requested), to conserve bandwidth through aggregation of received unicast replies from child nodes. To perform aggregation, parent relays wait a specific amount of time for a reply from child nodes before timing out. This delay is show in Equation 6.2. The $Timeout\_Constant$ and $Sink\_Degree$ are used to limit the length of the delay a relay waits for information from its child nodes, before aggregating replies and forwarding the aggregated information to its parent relay. The $Sink\_Degree$ is the number of hops from the sink a relay is. The $Timeout\_Constant$ needs to be selected based upon the expected sink degree and should be sufficiently large enough to ensure relays further away from the sink are able to delay sufficiently to collect and aggregate replies from their child nodes. A small $Timeout\_Constant$ will reduce the amount of aggregation of information by relays further from the sink, while those closer to the sink will perform more aggregation. The intent of this simple approach is too allow the flow of information back to the sink as a ”wave front”. Thus allowing aggregation of information at relays to reduce, where possible, the amount of duplicated information. We intend in future work to look at other mechanisms for controlling this timeout period, which may be determined dynamically by RAIC.

$$T_{out} = \frac{Timeout\_Constant}{Sink\_Degree}$$  \hspace{1cm} (6.2)

### 6.3 Simulation Environment

A simulation was developed to compare the performance of RAIC with both a Direct Response (Deb et al., 2002) mechanism and Directed Diffusion (Intanagonwiwat
et al., 2000) by initiating a query of the ad hoc network. Direct Response was chosen as it provides a brute force approach to sensing and is applicable to ad hoc networks where a routing protocol may be used to direct replies back to the sink. Directed Diffusion was chosen for comparison as it may be implemented using current ad hoc network networking technology and like RAIC is an aggregated sensing mechanism. In our implementation of a Direct Response mechanism, the sink initiates an optimised flood using MPR to disseminate a SREQ message. MPR is used as it is an optimised flooding mechanism that is comparable with UBF, thus reducing the overhead associated with disseminating the SREQ message. A reverse path back to the sink via the preceding node is maintained at each receiving node. A node upon receiving a SREQ message unicasts a SREP message back to the sink via the reverse path determined during the flood. The purpose of the query from the sink is to collect all node identifiers in the network. The simulation is executed multiple times with a different seed and the sink randomly selected. At the completion of a query the sink verifies the collected information (node identifiers) with the ideal obtainable information. The final results are averaged and 95% confidence intervals are generated.

The simulation generates a stationary random topology of nodes. Nodes have a maximum transmission range of 100 meters. Time is divided into epochs. An ideal MAC layer is assumed. There is no medium contention nor hidden-node scenario within the simulation as it is assumed that during an epoch all nodes can complete their respective transmission. The transmission medium is error free. A bidirectional link between two nodes is assumed upon reception of a beacon message. A first order radio model (Heinzelman et al., 2000) is assumed. In this model the first order radio dissipates $E_{elec} = 50nJ/bit$ to run the circuitry of a transmitter or receiver and a further $\epsilon_{amp} = 100pJ/(bit * m^2)$ for the transmitter amplifier. Equation 6.3 is used to calculate the costs of transmitting a $k$-bit message a distance $d$. Equation 6.4 is used to calculate the costs of receiving a $k$-bit message. The radios have power control and consume the minimal required energy to reach the intended recipients. The additional costs (RTS/CTS/ACK) associated with unicasting messages as opposed to broadcasting messages are accounted for.
\[ E_{T_x}(k, d) = E_{\text{elec}} \cdot k + \epsilon_{\text{amp}} \cdot k \cdot d^2 \]  

(6.3)

\[ E_{R_x}(k) = E_{\text{elec}} \cdot k \]  

(6.4)

Nodes in the network are assigned a random amount of remaining energy up to 2 joules. Additionally, each node has its own user-based constraints described earlier as benevolence (B). These constraints inhibit a node from rebroadcasting a SREQ message should the node’s remaining energy drop below a fixed level - referred to as "minimum energy". A minimum energy node is allowed to reply to a SREQ messages with its own SREP message, but not forward SREP messages received from other nodes. This is similar to a user with a laptop who wishes to use the network and support the services of the ad hoc network. However, a user may not wish the laptop’s battery supply to be fully depleted while supporting network services. Clearly, a user may desire to specify what services the laptop is able to support given specific battery levels of the laptop.

6.4 Simulation Results

Figures 6.3, 6.4 and 6.5 show the simulation results for RAIC, Direct Response and Directed Diffusion for 300 queries of a 100 node ad hoc network. Figure 6.3 shows the performance in terms of collected information for each of the mechanisms. RAIC is shown to provide significant performance improvement over multiple queries of an ad hoc network. RAIC after 300 successive queries of the ad hoc network is able to collect information from 85% of the network, compared with 24% for Directed Diffusion. Direct Response depletes the nodes in the network and therefore has significantly worse performance. Additionally RAIC is able to collect information from over 90% of the network up to 262 queries. This is quite significant compared to 28 queries for Directed Diffusion and 5 queries for Direct Response. Figures 6.4 and 6.5 show the number of nodes that enter the minimum energy state and those that become depleted due to excessive use. In Figure 6.4 at the completion of 300
Figure 6.3 A comparison of the percentage of nodes information is collected from over multiple successive queries.

Figure 6.4 A comparison of the number of nodes that enter a minimum energy state over multiple successive queries.
queries, the number of minimum energy nodes with RAIC is 43% less than with Directed Diffusion. In Figure 6.5 at the completion of all queries, RAIC has 38% less depleted nodes than Directed Diffusion. In the constrained environment described, Direct Response performs poorly as it is a brute force approach and therefore is not aware of the resources available, nor is it efficient at collecting information from the ad hoc network. In Figures 6.4 and 6.5, the initial growth (up to 40 queries) of the number of minimum energy nodes and depleted nodes is quite dramatic for Direct Response. The number of depleted nodes reaching 13 and minimum energy nodes reaching 53. At this point, the nodes in the ad hoc network are not able to support Direct Response which then fails to collect information from the ad hoc network. This accounts for the decreasing gradient for Direct Response in the graphs.

The use of blind flooding in directed diffusion may be beneficial in the setup phase in a sensor network as it is a brute force approach to disseminating a request for information and allows for the formation of multiple paths back to the sink. However, blind flooding is resource blind and cannot determine if a node has sufficient energy to relay information back to the sink. A recovery mechanism may be used to try
alternative paths, but this introduces additional overhead. The use of an optimised flooding mechanism in RAIC results in fewer paths back to the sink. However RAIC selects the most useful nodes during setup to be relays. During the capture phase, relays only unicast a reply to a parent relay that offers the best path back to the sink based upon the reverse path utility determined during setup. This greatly improves RAIC’s performance (Figure 6.3) and ability to query a constrained ad hoc network.

Figures 6.6, 6.7 and 6.8 show the simulation results for RAIC, Direct Response and Directed Diffusion for a single query of an ad hoc network. The concentration of the nodes is kept constant at 166.7 nodes per square kilometre. In a network consisting of 200 nodes, RAIC to perform a single query of the ad hoc network requires 90\% less energy, 84\% less packet transmissions and 88\% less packet receptions than Direct Response. Compared to Directed Diffusion, RAIC to perform a single query of the ad hoc network consumes 28\% less energy, transmits 29\% less packets and receives 35\% less packets. Thus RAIC provides significant performance improvement in terms of reducing the broadcast storm problem and energy consumption compared to Direct Response and Directed Diffusion. The aim of RAIC is to collect information from the ad hoc network without introducing significant overhead as this...
Figure 6.7 A comparison of the packets transmitted by nodes per single query of the ad hoc network.

Figure 6.8 A comparison of the packets received by nodes per single query of the ad hoc network.
overhead would reduce the overall capacity of the ad hoc network.

6.5 Conclusions

In this chapter we have introduced RAIC, a distributed, two phase (setup and capture), resource aware approach to information collection in ad hoc networks. A node selected as a sink initiates an optimized flood of a sensory request packet throughout the ad hoc network. This process of dissemination is resource aware, in that the relay nodes are selected based upon their ability to partake in both rebroadcasting the sensory request packet and also in forming a reliable tree-based sensory backbone during the capture phase. Upon receiving a sensory request packet, a node selected as relay then collects unicast replies from child nodes and aggregates responses before unicasting a reply to its parent relay. Nodes are able to select their parent relay based upon their distance and also upon their reverse path utility back to the sink node.

RAIC greatly reduces problems associated with information collection in wireless networks in terms of power consumption, the broadcast storm problem, implosion, resource blindness and overlap. We show through simulation that in terms of energy consumption, received and transmitted packets, RAIC performs significantly better than a Direct Response approach. Additionally compared to an aggregated response approach such as Directed Diffusion, RAIC shows significant performance improvement in terms of reducing the broadcast storm problem and energy consumption.

More importantly, we show that in an ad hoc network which is both energy constrained and has simple user constraints, RAIC is able to significantly extend the lifetime of the network while performing multiple queries. This is due to the resource awareness during the setup and capture phases of RAIC.
Chapter 7

Conclusions and Future Work

7.1 Overview

The advent of ad hoc networking based upon wireless technology is set to bring about a new revolution in how we communicate and share information. While many problems associated with communication in ad hoc networks have been covered by existing literature. This thesis has addressed some of the vital areas within ad hoc networking. Specifically, we have addressed issues relating to information dissemination such as: the use of transmission power control to further limit the broadcast storm problem; the necessity of resource awareness and the mechanisms to achieve resource awareness in constrained environments; the need for more reliable flooding mechanisms in the presence of background traffic. Furthermore, we have presented an approach to information collection that is suited to operation in ad hoc network environments and which builds upon work done in sensor networks and our work in resource aware information dissemination. In this chapter we provide a summary of the main ideas and findings presented in the preceding chapters. We also present the main conclusions drawn from the analysis carried out for the proposed mechanisms and results obtained. Finally, we present a discussion of the future work in this area.
7.2 Significant Results

In Chapter 2, we introduce the concepts, mechanisms and technologies that comprise ad hoc networking. We then review and categorise existing literature and discuss the problems associated with information dissemination and information collection mechanisms in ad hoc networks. The Literature review identifies the need for new information dissemination mechanisms that utilise transmission power control, are resource aware and provide reliable delivery of packets. Additionally we identify the need for information collection mechanisms that are suited to operation in ad hoc network environments.

Mechanisms for information dissemination are optimised to reduce the broadcast storm problem by reducing the number of redundant rebroadcasts. This is achieved by limiting the number of participating nodes. The approaches, as described in the literature review, with best performance to this problem for ad hoc network are distributed, use local neighbour knowledge (up to two hops) and are based upon some form of greedy set cover algorithm. Finding a reduced set of participating nodes based upon greedy set cover is an NP-complete problem. Therefore various heuristics have been proposed such as in Multipoint Relay flooding. These approaches do not scale well as their performance is limited by increasing node density and limited processing capabilities. Importantly the approaches described in the literature review only consider the use of a constant (full power) transmission range by all nodes when broadcasting. In Chapter 3, we have proposed two mechanisms: Neighbour Aware Adaptive Power (NAAP) flooding and Localised Minimum Spanning Tree Flooding (LMSTFlood). Both mechanisms allow for the transmission range of a broadcast to be adjusted to only include (where possible) those necessary nodes within a broadcast. This limits the effect of broadcasts on neighbouring nodes which may not need to receive a broadcast. The proposed mechanisms are shown to scale significantly better (in terms of reducing the broadcast storm problem and energy consumption) in high node densities than those optimised mechanisms based upon a constant transmission range.

In Chapter 4 we have proposed two optimised resource aware flooding mechanisms:
Utility based Multipoint Relay (UMPR) Flooding and Utility Based Flooding (UBF). Resource awareness implies that the mechanisms are able to make decisions about which nodes are selected to rebroadcast based upon available resources. These resources may be a node’s available battery power or constraints imposed upon a device's behaviour by a user. The approach of existing optimised flooding mechanisms is to select rebroadcasting nodes irrespective of available resources. Thus some nodes selected or assumed to rebroadcast may not, thereby inhibiting the propagation of a flood. Chapter 4 shows how the proposed mechanisms by accounting for available node resources have significantly improved performance in an ad hoc network environment that consists of nodes constrained by battery power and simple user based constraints. The proposed mechanisms are able to extend the lifetime of the network thereby allowing for repeated successive floods to reach above 90% of the nodes in the ad hoc network. Importantly, the addition of resource awareness in the proposed mechanisms did not impact overhead performance compared with existing optimised mechanisms.

Optimised flooding mechanisms as described in the literature review and those proposed in Chapter 3 reduce problems associated with blind flooding by limiting redundant broadcasts and reducing the effects of broadcasting by reducing transmission power. However, the very nature of Blind flooding and the significantly redundant transmissions ensures higher reliability than optimised flooding mechanisms. In Chapter 5, we have shown that the reliability of optimised flooding mechanism are drastically affected by background traffic compared to Blind flooding. This is due to the use of unreliable broadcasts being affected by unicast transmissions and other broadcast transmissions. Therefore, in Chapter 5, we have proposed Reliable Minimum Spanning Tree (RMST) flooding. RMST is an optimised flooding mechanism that takes advantage of the unique nature of the Minimum Spanning Tree (MST) to replace unreliable broadcast transmissions (as used by existing optimised flooding mechanisms) with more reliable unicast transmissions. Through simulation RMST is shown to have equivalent reliability to Blind flooding with overhead performance exceeding MPR and approaching that of LMSTFlood.

Information collection is the process of collecting information at a single node from
all nodes in the network. Essentially it is an all to one process which is the opposite of information dissemination. Information collection mechanisms rely upon information dissemination as means of disseminating requests for information and therefore are subject to similar problems. In sensor network literature there is much work on mechanisms for information collection. However, much of this work focuses on sensor networks, not on ad hoc networks. In Chapter 6, we have presented Resource Aware Information Collection (RAIC), a distributed information collection mechanism for ad hoc networks. RAIC makes it possible for a node in an ad hoc network acting as a sink to collect information from all other nodes within the network. RAIC takes advantage of the differences between ad hoc networks and sensor networks, mainly the existence of neighbour knowledge that allows for an optimised and resource aware flooding mechanism (UBF - Chapter 4) to be utilised. The use of UBF as the basic mechanism of disseminating a request for information allows RAIC to select only those nodes most suited as relays based upon their available resources. This is important as these relays also form a directed backbone that must be capable of relaying information back to the sink. As information flows back to the sink, it is possible for nodes to collect and collate the information to reduce the number of packets flowing back to the sink. The path that packets follow towards the sink is based upon a reverse path utility. The reverse path utility allows packets to be routed via those relays that provide the best path in terms of their remaining battery power and relay density. RAIC is compared to Directed Diffusion and a brute force approach (Direct Response) and shown to be able to collect information from nodes in the ad hoc network with less overhead in terms of packets transmitted and received, and energy consumption. Additionally given an ad hoc network with resource constrained nodes, RAIC is able to perform significantly more repeated requests for information than Directed Diffusion or Direct Response.

7.3 Further Work

This thesis has addressed a number of significant issues associated with information dissemination and information collection. However, there are still a number of issues that require further investigation. These are described below:
• The optimised flooding mechanisms described in Chapter 3, Chapter 4 and Chapter 5 require knowledge of local neighbouring nodes either one or two hop away. This is achieved through the exchange of periodic beacon messages or passively through background traffic. This knowledge is used to make rebroadcast decisions. Future work should consider the effects of mobility (and degree of mobility) upon the knowledge that is used for rebroadcast decisions and how it affects optimised flooding mechanisms. Additionally what mechanisms may be utilised to counteract the effects due to incorrect or old neighbour knowledge.

• In Chapter 4, we have introduced two optimised resource aware flooding mechanisms. However, these mechanisms do not adjust transmission power when broadcasting as is done in Chapter 3. Thus they will not scale well (in terms of the broadcast storm problem and energy consumption) in high node densities as do the optimised flooding mechanisms (Chapter 3) that utilise transmission power control. Thus, there exists potential for more work in resource aware flooding that improves upon the the proposed mechanisms at reducing the broadcast storm problem.

• To date there has been little work on the effects of optimised flooding mechanisms on routing protocols. Blind flooding may be used in AODV to reactively discover routes between a source and destination node. In this case, how would an optimised flooding mechanisms perform? An optimised flooding mechanism would definitely reduce overhead associated with route discovery which is desirable. However, the paths along which an optimised flood may propagate may be not necessarily be suitable as a path between a source and destination node. As is shown in Chapter 3, optimised flooding mechanisms have varying resulting characteristics that may affect route discovery and the usefulness of any resulting path.

• In Chapter 5, we presented an optimised flooding mechanism that provides more reliable packet delivery than optimised flooding mechanisms. This approach is different from the reliable flooding mechanisms described in the literature review. The reason for this is that RMST improves reliability on a per
hop basis by utilising more reliable unicast packet transmission as opposed to unreliable broadcast transmission. However, there is no guarantee of delivery and no means for a node initiating a flood to discover which nodes received the flood. Reliability will play an important role in ad hoc networking and it is therefore necessary that mechanisms other than those described in the literature review be researched because the reliable mechanisms proposed in literature are not optimised to reduce the broadcast storm problem.

- In proactive routing protocols, nodes must flood link state information when changes are detected. Given a network of sufficient size and mobility this dissemination will begin to consume large amounts of available bandwidth. A possible area of further work is in all to all flooding mechanisms. In an all to all flooding approach, it is assumed that there exists multiple floods disseminating link state information simultaneously. An open research question is: is it possible to unify these separate floods thereby fusing their combined data. Floods may therefore be unified where possible to reduce overhead. It may also be possible to create and maintain efficient backbones of relay nodes throughout the ad hoc network to facilitate the fusion of floods.

- Hierarchal information dissemination mechanisms may be useful in limiting the scope of information dissemination in ad hoc networks. Additionally as the size of an ad hoc network grows, hierarchal mechanisms may be necessary for scalability purposes.

- In heterogenous ad hoc network environments, nodes may have varying transmission ranges as they may have different communication hardware and constraints. Flooding mechanisms that are able to utilise the heterogeneous broadcast ranges may be beneficial to route discovery mechanisms such as AODV where an expanding ring approach is taken to route discovery. Thus nodes with higher power and larger broadcast ranges are utilised before those with lower broadcast range and lower power.

- Ad hoc networks that consist of some physical network infrastructure may require flooding mechanisms that are able to take advantage of the physical infrastructure. Nodes connected to a physical infrastructure may disseminate
information between each other without the wireless overheads associated with flooding. Nodes connected to the physical infrastructure may then "inject" packets being flooded back into the ad hoc network at specific points.

- When collecting information from an ad hoc network, reliability is an important issue. It is therefore important that future work in RAIC should consider the effects of mobility and node failure, thus the performance of RAIC in mobile scenarios should be determined and mechanisms for recovery and the reliable collection of information should be addressed.

- The optimised flooding mechanisms described in this thesis provide a heuristic based solution to improving the performance of flooding in a distributed manner. Further analytical work could develop techniques for determining global optimums of some of the algorithms developed in this thesis. For example: minimizing a combined cost which includes the energy consumed due to packet transmission and reception during flooding as well as the utilities defined in this thesis. Global optimisation is not in itself useful in ad hoc environments, however, it would allow us to determine how effective our distributed approaches are.
Bibliography


delivery in mobile ad hoc networks. In Mobile Networks and Applications,
pages vol. 4, pp. 175–192.

139.

Networks, pages vol. 3, pp. 143–156.

ad hoc networks. In Proceedings of the first ACM international symposium on
mobile and ad hoc networking and computing, Boston, Massachusetts, USA,

ad hoc networks. In Journal of Science and Technology - Beijing, China.

Perkins, C. and Bhagwat, P. (1994). Highly Dynamic destination-sequenced dis-
tance vector routing (DSDV) for mobile computers. In Proceedings Computer
Communications Review, pages pp. 234–244.

100.


Appendix A

Verification of Simulations

In this thesis we have obtained simulation results for optimised flooding mechanisms based upon a simplified simulation environment and GlomoSim (Bajaj et al., 1999). The simplified simulation environment allows us to abstract the underlying inherent complexity, behaviour and implementation of the information dissemination and collection mechanisms from problems associated with a specific protocol stack and MAC layer. This allows us to more accurately determine the actual behaviour and performance of the proposed mechanisms.

Simulation results from Chapters 3, 4 and 6 are obtained using a simplified simulation environment. In the simplified simulation environment, time is divided into epochs. An ideal MAC layer is assumed, which removes problems associated with medium contention and hidden-node scenarios. Thus it is assumed that during an epoch all nodes may safely transmit packets without collision or error and the packets received by all nodes within the maximum transmission range of the node. In terms of information dissemination, this simplification allows for more accurate determination of the actual behaviour and performance of optimised flooding mechanisms at limiting the Broadcast Storm Problem (Ni et al., 1999), without worrying about implementation issues and MAC layer issues (which may vary depending upon the specification).

To further test proposed approaches in non-ideal environments the GlomoSim simulation environment was used. GlomoSim is a parallel simulation environment im-
implemented in PARSEC, PARallel Simulation Environment for Complex Systems (UCLA, 2004). GlomoSim provides a layered structure that includes radio propagation, medium access control (MAC), link layer, network layer, transport layer and application layer implementations. In addition, various wireless and ad hoc network protocols are already implemented. GlomoSim allows for detailed modelling of several layers and the study of their interaction, while preserving very good runtime efficiency. In Chapter 5, we consider the effects of more realistic propagation environment based upon the IEEE 802.11 MAC layer and verify how optimised flooding is affected. We use constant bit rate (CBR) traffic sources to inject traffic into the ad hoc network. This CBR traffic effectively creates noise in the network, thus stressing the performance of the optimised flooding mechanism and allowing for its actual performance in terms of reliability to be tested.

As we have chosen to use both a simplified simulation environment and GlomoSim. It is important to verify these environments against each other. We also further verify against NS-2 (NS-2, 2004). The Multipoint Relay (MPR) (Qayyum et al., 2001) optimised flooding algorithm is used as a base mechanism for comparison, as it is available in both GlomoSim and NS2 where it is implemented as part of the Optimised Link State Routing Protocol (OLSR) (Jacquet et al., 2000). Importantly, MPR is used in published literature as a base optimised flooding mechanism against which to compare performance of proposed mechanisms. Figures A.1 and A.2 show simulation results for various implementations of MPR as implemented in GlomoSim and NS-2. Furthermore, we provide simulation results for our own implementation of MPR in the simplified simulation environment and in GlomoSim. From Figures A.1 and A.2, we can see that all four simulations show equivalent transmitted and received packets for a single optimised MPR flood. The slight difference in simulation results is possibly due to the different underlying implementations. Both NS-2 and GlomoSim have IEEE 802.11 implementations, however, slightly different MAC and propagation models are used, which result in different simulation results as confirmed in (Royer et al., 2000). However, all results are within range of each other and those obtained with the ideal MAC as used by the simplified simulation environment.
Verification of Simulations

Figure A.1 Comparison of transmitted packets for a single MPR flood in Glomosim, NS2 and Simplified simulation

Figure A.2 Comparison of received packets for a single MPR flood in Glomosim, NS2 and Simplified simulation
Appendix B

Notes on Simulation Randomness, Correlation and Confidence

In (Pawlikowski et al., 2002), the authors state that a generally accepted and commonly used approach in simulation environments is to use algorithmic generators of uniformly distributed pseudo-random numbers as basic sources of randomness in stochastic simulations. This results in a periodic sequence of numbers as all numbers are determined at the start of execution given a specific seed value and thus querying a pseudo-random number generator may result in a cyclic use of the same numbers. This is problematic in ad hoc network simulation environments where the length of a simulation run may be large and thus exhaust the supply of unique pseudo-random numbers. This may lead to correlation between simulation runs. To avoid this problem of correlation between simulation runs, it is suggested to vary the starting seed value of each simulation run and to execute simulation runs independent of prior simulation runs. In this thesis, all simulation runs are initiated from the start with a new incremental seed value, thus ensuring each simulation run has a different seed value to the previous simulation run. More importantly, each simulation run lasts only as long as it takes to complete a flood in the case of the information dissemination mechanisms or in the case of the information collection mechanisms to sense the state of the ad hoc network before a timer expires.

The use of a different seed value for each simulation run has a specific effect upon:
• The resulting ad hoc network topology used by both simulation environments varies for each simulation run.

• In information dissemination simulations, the selection of the starting node that initiates a flood is determined randomly and thus changes for each simulation run.

• In information collection simulations, the selection of the sink node is selected randomly and thus differs in each simulation run.

• In GlomoSim, the random seed also varies the resulting effects of physical layer packet loss due to MAC level frame collisions and the selection of CBR source destination pairs used to generate network traffic.

In (Pawlikowski et al., 2002) the authors state that stochastic simulations must be regarded as simulated statistical experiments and thus application of statistical methods of analysis of simulation results is necessary. It is therefore important to calculate the degree of confidence in the accuracy of collected simulation results. This degree of accuracy is referred to as a confidence interval (in this thesis 95 percent). The authors state that in a correctly implemented simulation environment, the collected results should exhibit confidence intervals that decrease the longer the simulation environment is run. Two approaches exist for determining the duration of a simulation. The first approach, sequential scenario simulation is to run the simulation for an undetermined period of time and to consecutively check the accuracy of the estimates obtained. The simulation is stopped when the relative error of the estimates falls below an acceptable threshold. The second approach as used in this thesis is the finite time horizon simulation. This approach requires that each simulation test a specified period of time or process length (for example one complete flood of an ad hoc network or one complete sense of an ad hoc network). In finite time horizon simulations, it is important that the simulation environment is run an appropriate number of times with a different statistically independent sequence of pseudo-random numbers as sources of randomness. In (Pawlikowski et al., 2002), the authors state that this ensures that the sample of collected data from the simulation is representative of independent and identically distributed random variables. Thus, allowing for 95 per-
cent confidence intervals to be calculated using standardised approaches. In Chapter 3 to Chapter 6, all simulations in both GlomoSim and the simplified event simulator are executed exactly 100 times. The final results of all simulation runs are averaged and 95 percent confidence intervals are calculated as defined in Appendix C of (Fishman, 1978).
Appendix C

Ad hoc Network Topology Generation

In this thesis all ad hoc network topologies are randomly created using Algorithm *plot random topology*(). This simple algorithm is used in GlomoSim (Bajaj et al., 1999) and NS-2 (NS-2, 2004). Each simulation run is provided with a different random seed value to the previous simulation run, thus each simulation run results in a different ad hoc network topology.

**Algorithm** *plot random topology*()

1. for each node i
2. \[\text{node}[i].x \leftarrow \text{random(seed)} \times \text{maxX}\]
3. \[\text{node}[i].y \leftarrow \text{random(seed)} \times \text{maxY}\]
4. endfor

One issue with this approach of randomly plotting an ad hoc network topology is that it is difficult to ensure that the resulting network topology is connected, especially at low node densities. It is possible to only select those random topologies that are connected, however this is time consuming. It is also possible to use a uniform distribution of nodes to create a uniform ad hoc network topology. However, uniformly distributed topologies are considered to be not realistic in ad hoc network literature. Therefore in this thesis, we determine the total number of nodes reachable in a partitioned random topology from the source node of a flood (or sink node for information collection) given each node’s full transmission range. The collected results are then
calculated with the total number of nodes in the partition of the source node. As the node density increases, the problem of a partitioned topology is reduced. In information dissemination and information collection, we are more concerned with dense ad hoc network topologies. Thus, those topologies with low node density are not as important. Hence, this approach to topology generation is sufficient.

In this thesis all simulations generate an ad hoc network topology in an area 600 meters by 600 meters. Node's are assigned a maximum transmission range of 100 meters. Figure C.1 shows the resulting average node degree for increasing node density from 40 nodes up to 200 nodes.