ADVANCED ROUTING TECHNIQUES FOR WIRELESS AD HOC NETWORKS

TAN CHEE WAH

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University in fulfillment of the requirement for the degree of Doctor of Philosophy

2009
Acknowledgements

I would like to take this opportunity to thank the following people who have given me their support throughout my Doctor of Philosophy (PhD) candidature. I am grateful to my wife, Ms. Teo Ning Hui, and my family members for which, without their strongest support and deepest understanding, I would not be able to embark on the PhD path. Next, I would like to thank my supervisors Assoc. Prof. Sanjay K. Bose for his continuous guidance and advice given, and Assoc. Prof. Cheng Tee Hiang for his insightful comments and help along the way. I would also like to thank Assoc. Prof. Law Choi Look, Assoc. Prof. Guan Yong Liang, Mr. Ng Teng Kwee, Ms. Ong Pei See, Ms. Chay Ooi Mei and all other technical officers from the Positioning and Wireless Technology Centre (PWTC) for providing a conducive environment to carry out my research activities. I appreciate the friendship and assistance rendered by PWTC research staff and students, especially Mr. Seow Chee Kiat, Dr. Lum Kum Meng, Mr. Wang Wei, Mr. Yogesh Pandit, Mr. Raghu Kumbakonam Ramadurai and Mr. Tian Hongtao. Last, but not least, I would like to thank all the people who have helped me in one way or another to complete my project and thesis.
# Table of Contents

Acknowledgments i  
Summary vi  
List of Figures ix  
List of Tables xiii  
List of Acronyms xiv

## Chapter 1 Introduction 1

1.1 Motivation ......................................................... 1  
1.2 Objectives and Contributions of the Thesis .................. 7  
1.3 Organization of the Thesis .................................... 9

## Chapter 2 Literature Review 12

2.1 On-demand Routing ............................................. 12  
2.1.1 Ad hoc On-demand Distance Vector Routing ............ 12  
2.1.2 Dynamic Source Routing .................................. 15  
2.1.3 Protocol Comparison ...................................... 17  
2.1.4 Performance Comparison ................................. 19  
2.2 Power Aware Protocols ....................................... 23  
2.2.1 MAC Layer Power Aware Protocols ...................... 23  
2.2.2 Network Layer Power Aware Protocols .................. 28  
2.3 Cooperation Enforcement Techniques ........................ 38  
2.3.1 Reputation-based Systems ............................... 39  
2.3.2 Behavioral-based Systems ............................... 41  
2.3.3 Reward-based Systems ................................. 43
# Table of Contents

## 2.4 Multi-objective Routing

### Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

- 3.1 Introduction .................................................. 54
- 3.2 Protocol Description ........................................ 55
- 3.3 Route Discovery .............................................. 56
  - 3.3.1 Reverse Path Setup ................................... 58
  - 3.3.2 Forward Path Setup .................................. 59
- 3.4 Route Maintenance .......................................... 62
- 3.5 Cost Maintenance ........................................... 64
  - 3.5.1 Basic Concept ......................................... 64
  - 3.5.2 Practical Design ..................................... 65
- 3.6 Protocol Comparison ........................................ 69
- 3.7 Performance Comparison ................................. 73
  - 3.7.1 Simulation Environment ............................... 74
  - 3.7.2 Stationary Scenario Results and Discussion ........ 77
  - 3.7.3 Mobile Scenario Results and Discussion .......... 80
  - 3.7.4 System Performance with Hop Count Limit Imposed . 84
- 3.8 Conclusion ..................................................... 90

## Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

- 4.1 Introduction .................................................. 91
- 4.2 Cost-Credit based Forwarding ............................ 92
  - 4.2.1 A Packet Forwarding Strategy in a Simple System Model... 92
  - 4.2.2 Operation in a Simple System ......................... 94
  - 4.2.3 Operation in a Real System .......................... 96
- 4.3 On-demand Cost-Credit Routing .......................... 98
4.3.1 Route Discovery ................................................. 98
4.3.2 Route Management ........................................... 103

4.4 Performance Studies ........................................... 104
4.4.1 Static Network Results and Discussions .......... 104
4.4.2 Mobile Network Results and Discussions ......... 118

4.5 Centralized Power Credit Routing ......................... 121
4.5.1 General Network Lifetime Formulation ............. 121
4.5.2 Max-min Power Credit Routing ....................... 125

4.6 Performance Comparison of Two Cost-Credit based Routing Strategies ....................... 130

4.7 Conclusion ....................................................... 135

Chapter 5 Location-aided Multi-Objective Routing 137

5.1 Introduction ..................................................... 137

5.2 Power Aware Multi-Objective Routing ................. 138
5.2.1 General Problem Formulation ......................... 138
5.2.2 Variations with Load-Dependent Formulations .... 140
5.2.3 Variation with Known Transmission Requirement 142

5.3 On-demand Location-aided Power Aware Routing ...... 144
5.3.1 Flood Control Mechanism .............................. 144
5.3.2 Route Request Phase ..................................... 149
5.3.3 Route Reply Phase ........................................ 162
5.3.4 Performance Studies in Static Network ............. 164

5.4 Power and Mobility Aware Multi-Objective Routing ... 170
5.4.1 Problem Formulation ...................................... 170
5.4.2 Variation with Known Transmission Requirement 174
## Table of Contents

5.5 On-demand Location-aided Power and Mobility Aware Routing… 175

5.5.1 Route Discovery......................................................... 175

5.5.2 Route Maintenance..................................................... 185

5.5.3 Performance Evaluation in Mobile Network............... 185

5.6 Conclusion...................................................................... 193

Chapter 6 Conclusion and Recommendations 196

6.1 Conclusion...................................................................... 196

6.2 Recommendations for Further Research......................... 200

Author’s Publications .......................................................... 202

Bibliography ...................................................................... 203

Appendix A Derivation of Initial Route Cost Value for ODCCR 215
Summary

The pervasiveness of wireless communications and the need to connect “anyplace, anytime, anywhere” has led to the development of wireless ad hoc networks. While the current IEEE 802.11 standard allows wireless communications to be established in a LAN (local area network) like manner, its support for ad hoc communications is limited. This is because the IEEE 802.11 standard requires an established infrastructure to be in place for routing and other networking tasks. Wireless devices which move out of radio communication range of this network infrastructure cannot retain their network connectivity. In contrast, in a wireless ad hoc network, these devices may still be able to communicate with each other as long as each device is within the communication range of other devices, even if there is no established network infrastructure in place. This is achieved by having all the wireless devices perform additional functions of data forwarding and routing to support wireless communications, apart from being the eventual source and destination of the data.

Several substantially open issues which may be identified with regard to wireless ad hoc networking are addressed here. Since wireless devices or nodes are typically mobile, they may be run on a limited battery power supply and may not always be connected to a power outlet for replenishment. In view of this, the additional routing workload should not consume too much of a node’s limited battery power resource. To address this issue, an on-demand routing protocol, Power Aware Ad hoc On-Demand Distance Vector routing protocol (PAW-AODV), that uses the limited power of the wireless nodes efficiently is proposed. PAW-AODV has been devised by proposing appropriate power awareness features which are then integrated into the standard AODV protocol. PAW-
Summary

AODV still retains the basic route discovery and maintenance mechanism of AODV. Performance studies conducted show that PAW-AODV can achieve better throughput performance due to its power aware algorithm and that it results in nodes staying activated for a longer period of time. This is done without causing significant degradation to its delay performance.

Another issue is the willingness of the nodes in altruistically parting with their own power resources for routing transmissions of other nodes. It would be reasonable to expect that nodes will cooperate only if they find that this will not adversely affect their own transmissions significantly. An On-Demand Cost-Credit Routing (ODCCR) protocol for cooperative routing using a cost-credit concept is proposed to address this and an AODV-like implementation for this has also been suggested. A forwarding rule is proposed and deployed in ODCCR to nominally allow nodes to increase their self-transmissions in this cooperative and power constrained environment. A centralized routing protocol based on maximizing the minimum battery power among all nodes for the cost-credit system is also proposed. Routes selected must enable source nodes to transmit as many data packets as possible. Performance studies conducted show that both protocols perform well in terms of network lifetimes and throughput performance.

Node mobility is another major concern in mobile ad-hoc networks since this (in conjunction with the limited battery power of the node) will have an adverse effect on both route establishment and route maintenance. With modern technology (e.g. GPS and other localization approaches), location information may be assumed to be available to each node. In turn, this information may be used to compute routing parameters like transmission power and route lifetime which can then be incorporated in a multi-
objective on-demand routing approach. In the simpler case of a static network, route
selection following this approach will be based on both maximizing the minimum node
battery power and minimizing the total transmission power required. In a mobile
network, route selection will additionally include maximizing the route lifetime as well.
Various on-demand multi-objective routing protocols are designed for real-time
operations in practical networks. One feature proposed for these routing protocols is a
mechanism to restrict control traffic flooding with the location information provided.
The second feature is a motion prediction module for computing link lifetime, which is
required to obtain route lifetime. The third feature is the use of the derived route lifetime
to pre-empt link breakage. Performance studies show that the proposed multi-objective
on-demand routing protocols are able to extend the time for which the networks can
operate and are also effective in restricting control traffic flooding.
List of Figures

Figure 1.1: Comparison of Two Types of Wireless Networks ......................... 2
Figure 1.2: Categorization of Routing Protocols for Wireless Ad hoc Networks (extracted from [1])................................................ 4
Figure 2.1: Route Discovery in AODV..................................................... 14
Figure 2.2: Route Discovery in DSR...................................................... 16
Figure 2.3: Packet Delivery Fraction and Average Data Packet Delays for a 100-node Network with Different Numbers of Sources (extracted from [38])................................................ 21
Figure 2.4: Performance with Increasing Offered Load for 100 nodes with 10 and 40 Sources (extracted from [38])................................................ 22
Figure 2.5: Number of Dead Nodes in DSR and PSR versus Time (extracted from [49])................................................................. 33
Figure 2.6: Operational cycle of a CPC Non-Backbone Node ..................... 38
Figure 2.7: Trust Architecture for CONFIDANT .................................... 40
Figure 2.8: Forwarding Rules for the Nuglet Currency System .................. 44
Figure 2.9: Message Success Rate vs. Network Battery Resource (extracted from [74])................................................................. 46
Figure 2.10: Power-Cost Localized Routing Algorithm .......................... 49
Figure 2.11: Determining Rebroadcast Delay in PANDA-TP...................... 51
Figure 2.12: Graph of Network Lifetime against Mobility Ratio (extracted from [77])................................................................. 52
Figure 3.1: RREQ Handling in PAW-AODV........................................ 58
Figure 3.2: Route Cost Computation in PAW-AODV .............................. 62
## List of Figures

| Figure 3.3: | RRCA Propagation and Handling in PAW-AODV | 65 |
| Figure 3.4: | Step-wise relationship between node cost and remaining battery power | 67 |
| Figure 3.5: | Process flow of AODV and PAW-AODV Route Discovery | 70 |
| Figure 3.6: | Process flow of AODV and PAW-AODV Route (Cost) Maintenance | 71 |
| Figure 3.7: | Node Placement for Stationary Scenario | 75 |
| Figure 3.8: | Performance Comparison of AODV and PAW-AODV in Stationary Scenario | 78 |
| Figure 3.9: | Performance Comparison of AODV and PAW-AODV in Mobile Scenario | 81 |
| Figure 3.10: | Graphs of Number of Packets Received against Time with the various Pause Times set | 83 |
| Figure 3.11: | Performance Comparison of AODV-ND4 and PAW-AODV-ND4 in a Stationary Scenario | 86 |
| Figure 3.12: | Performance Comparison of AODV-ND4 and PAW-AODV-ND4 in a Mobile Scenario | 88 |
| Figure 3.13: | Graphs of Number of Packets Received against Time with the various Hop Count Limit values set | 89 |
| Figure 4.1: | Graph of payout function w(t) against N_F | 96 |
| Figure 4.2: | Route Discovery Process of ODCCR | 102 |
| Figure 4.3: | Comparison of Network Lifetimes | 106 |
| Figure 4.4: | Comparison of Number of Packets Transmitted | 107 |
| Figure 4.5: | Comparison of Number of Packets Carried | 108 |
| Figure 4.6: | Network Lifetime Performance Comparison | 112 |
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4.7</td>
<td>Average Number of Data Packets Transmitted per Node within the Operational Network Lifetime</td>
<td>113</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Average Number of Data Packets Carried per Node within the Operational Network Lifetime</td>
<td>114</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Proportion of total data packets transmitted that are packets forwarded for others when ( C_f/B ) varies in the case of ODCCR</td>
<td>117</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Performance Comparison in Mobile Scenarios</td>
<td>120</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Overview of the Operation of max-min Power Credit Routing</td>
<td>126</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Pre-processing for maxminPowerOpt</td>
<td>127</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Routing and Post-processing after execution of maxminPowerOpt</td>
<td>129</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Performance Comparison of mmPCR, ODCCR and AODV</td>
<td>131</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Ratio of the number of multi-hop packets to the total amount of packets transmitted within the network lifetime in the case of ODCCR</td>
<td>135</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Computation to obtain the Forwarding Zone</td>
<td>148</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Forwarding and Destination Zone</td>
<td>149</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Illustration of positive and negative progress</td>
<td>149</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Route Request Process for LAMOR-PAL</td>
<td>153</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Route Request Process for LAMOR-PAF</td>
<td>154</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Route Request Process for LAMOR-PA/TR</td>
<td>155</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Check Constraints Module for LAMOR-PA/TR</td>
<td>159</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Routes Building Module in Destination Node for LAMOR-PA/TR</td>
<td>161</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Route Reply Process</td>
<td>164</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Comparison of LAMOR-PA/TR((Opt)) and LAMOR-PA/TR</td>
<td>166</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>Comparison of LAMOR-PAL((Opt)) and LAMOR-PAL</td>
<td>167</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.12</td>
<td>Comparison of (\text{LAMOR-PAF(Opt)}) and (\text{LAMOR-PAF})</td>
<td>167</td>
</tr>
<tr>
<td>5.13</td>
<td>Performance Comparison of (\text{LAMOR-PA}) protocols against (\text{HEAP}) and (\text{PANDA})</td>
<td>168</td>
</tr>
<tr>
<td>5.14</td>
<td>Diagram for Link Lifetime Derivation</td>
<td>171</td>
</tr>
<tr>
<td>5.15</td>
<td>Motion Prediction Module</td>
<td>177</td>
</tr>
<tr>
<td>5.16</td>
<td>Route Request Process for (\text{LAMOR-PMAMO}) and (\text{LAMOR-PMALF})</td>
<td>182</td>
</tr>
<tr>
<td>5.17</td>
<td>Route Request Process for (\text{LAMOR-PMA/TR})</td>
<td>183</td>
</tr>
<tr>
<td>5.18</td>
<td>Check Constraints Module for (\text{LAMOR-PMA/TR})</td>
<td>184</td>
</tr>
<tr>
<td>5.19</td>
<td>Graph of Network Operational Time against Pause Time</td>
<td>187</td>
</tr>
<tr>
<td>5.20</td>
<td>Graph of Network Lifetime against Pause Time</td>
<td>188</td>
</tr>
<tr>
<td>5.21</td>
<td>Graph of Number of Data Packets Carried within Network Lifetime against Pause Time</td>
<td>188</td>
</tr>
<tr>
<td>5.22</td>
<td>Graph of Energy Usage against Pause Time</td>
<td>190</td>
</tr>
<tr>
<td>5.23</td>
<td>Graph of Average Number of Route Error Packets (RERR) generated per Connection against Pause Time</td>
<td>191</td>
</tr>
<tr>
<td>5.24</td>
<td>Graph of the Ratio of Number of RREQs Forwarded to the Number of RREQs Initiated against Pause Time</td>
<td>193</td>
</tr>
<tr>
<td>A.1</td>
<td>Illustration of a two-hop connection scenario</td>
<td>217</td>
</tr>
</tbody>
</table>
List of Tables

Table 2.1: Differences between AODV and DSR ................................. 23
Table 3.1: Cost Zone Table .......................................................... 66
Table 3.2: Differences between AODV and PAW-AODV .................. 73
Table 3.3: Simulation Parameters.................................................... 74
Table 4.1: Values of the Parameters used ......................................... 109
Table 5.1: Definition of the various fields in RREQ .......................... 151
Table 5.2: Definition of the various fields in RREQ_Table ................. 152
Table 5.3: Definition of the various fields in RREP ........................... 163
Table 5.4: Definition of the various fields in Route Table .................... 164
Table 5.5: Definition of the additional fields in RREQ for LAMOR-PMA protocols ......................................................... 178
Table 5.6: Definition of the additional fields in RREQ_Table for LAMOR-PMA protocols ......................................................... 178
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABR</td>
<td>Associativity-Based Routing Protocol</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On-demand Distance Vector Routing Protocol</td>
</tr>
<tr>
<td>APTS</td>
<td>Acceptable-Power-to-Send</td>
</tr>
<tr>
<td>BAMAC</td>
<td>Battery Aware Medium Access Control</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>BU</td>
<td>Battery Unit</td>
</tr>
<tr>
<td>CCU</td>
<td>Cost-Credit Unit</td>
</tr>
<tr>
<td>CEDAR</td>
<td>Core Extraction Distributed Ad Hoc Routing Protocol</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CGSR</td>
<td>Cluster-Head Gateway Switch Routing Protocol</td>
</tr>
<tr>
<td>CONFIDANT</td>
<td>Cooperation of Nodes: Fairness in Dynamic Ad-hoc Networks</td>
</tr>
<tr>
<td>CORE</td>
<td>A Collaborative Reputation mechanism to enforce node cooperation in mobile ad hoc networks</td>
</tr>
<tr>
<td>CORSAC</td>
<td>Cooperation-optimal Routing-and-forwarding protocol</td>
</tr>
<tr>
<td>CPC</td>
<td>Coordinated Power Conservation</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear-to-send</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DIFS</td>
<td>DCF-IFS</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination Sequenced Distance-Vector Routing Protocol</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing Protocol</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FORP</td>
<td>Flow-Oriented Routing Protocol</td>
</tr>
</tbody>
</table>
### List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR</td>
<td>Forwarding Rule</td>
</tr>
<tr>
<td>FSR</td>
<td>Fisheye State Routing Protocol</td>
</tr>
<tr>
<td>GAF</td>
<td>Geographical Adaptive Fidelity</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSR</td>
<td>Global State Routing Protocol</td>
</tr>
<tr>
<td>GTFT</td>
<td>Generous Tit-for-Tat algorithm</td>
</tr>
<tr>
<td>HEAP</td>
<td>Hybrid adaptive Energy-Aware Routing</td>
</tr>
<tr>
<td>HSR</td>
<td>Hierarchical State Routing Protocol</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IFS</td>
<td>Inter-frame Spacing</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LAMOR</td>
<td>Location aided Multi-objective Routing</td>
</tr>
<tr>
<td>LAMOR-PA</td>
<td>LAMOR (Power Aware version)</td>
</tr>
<tr>
<td>LAMOR-PMA</td>
<td>LAMOR (Power and Mobility Aware version)</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LAR</td>
<td>Location-Aided Routing Protocol</td>
</tr>
<tr>
<td>LBR</td>
<td>Link Life Based Routing Protocol</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>mmPCR</td>
<td>max-min Power Credit Routing</td>
</tr>
<tr>
<td>ODCCR</td>
<td>On-Demand Cost-credit routing</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiple Access</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimized Link State Routing Protocol</td>
</tr>
<tr>
<td>PAMAS</td>
<td>Power Aware Multiple Access Scheme</td>
</tr>
<tr>
<td>PANDA</td>
<td>Positional Attribute based Next-hop Determination Approach</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>PAR</td>
<td>Power-Aware Routing Protocol</td>
</tr>
<tr>
<td>PARS</td>
<td>Power Aware Reputation System</td>
</tr>
<tr>
<td>PAW-AODV</td>
<td>Power Aware Ad hoc On-demand Distance Vector</td>
</tr>
<tr>
<td>PAW-RERR</td>
<td>Power Aware Route Error packet</td>
</tr>
<tr>
<td>PCMA</td>
<td>Power Control Multiple Access</td>
</tr>
<tr>
<td>PLBR</td>
<td>Preferred Link-Based Routing Protocol</td>
</tr>
<tr>
<td>PSR</td>
<td>Power-aware Source Routing</td>
</tr>
<tr>
<td>RABR</td>
<td>Route-Lifetime Assessment Based Routing Protocol</td>
</tr>
<tr>
<td>RBODR</td>
<td>Reward-based On-demand Routing</td>
</tr>
<tr>
<td>RERR</td>
<td>Route Error packet</td>
</tr>
<tr>
<td>RPTS</td>
<td>Request-Power-to-Send</td>
</tr>
<tr>
<td>RRCA</td>
<td>Route Cost Announcement packet</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply packet</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request packet</td>
</tr>
<tr>
<td>RTS</td>
<td>Request-to-send</td>
</tr>
<tr>
<td>SSA</td>
<td>Signal Stability-Based Adaptive Routing Protocol</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short IFS</td>
</tr>
<tr>
<td>SPRITE</td>
<td>Simple, Cheat-Proof, Credit-based System for Mobile Ad-hoc Networks</td>
</tr>
<tr>
<td>STAR</td>
<td>Source-Tree Adaptive Routing Protocol</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TFT</td>
<td>Tit-for-Tat strategy</td>
</tr>
<tr>
<td>TR</td>
<td>Transmission Rule</td>
</tr>
<tr>
<td>TTL</td>
<td>Time-To-Live</td>
</tr>
<tr>
<td>VDBP</td>
<td>Virtual Dynamic Backbone Protocol</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WRP</td>
<td>Wireless Routing Protocol</td>
</tr>
<tr>
<td>ZHLS</td>
<td>Zone-Based Hierarchical Link State Routing Protocol</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

1.1 Motivation

Wireless devices and services are becoming increasingly important, both during work and at home. Cell phones are used to call and send messages to one another. Palm devices and laptops are used to send emails and files via wireless networks inside or outside the workplace. Handheld devices are used to play online games with others using wireless interconnections. The need to connect “anyplace, anytime, anywhere” with ubiquitous connectivity is becoming increasingly important along with the need to always stay connected even without having any fixed network infrastructure in place. This is driving the need to establish and operate wireless ad-hoc networks for providing services even in situations which not only do not have any established infrastructure but also where the nodes may also be mobile in nature.

Wireless ad hoc networks, as shown in Figure 1.1(a), enable the wireless devices (which are likely to be mobile) to communicate on the fly without the presence of an established network infrastructure [1]-[3]. Unlike an infrastructure based wireless network as shown in Figure 1.1(b) (with an access point for wireless nodes to communicate with each other), a wireless ad hoc network does not need an access point. Most current wireless local area networks (LANs) based on the IEEE 802.11 standard [4]-[5] provide an infrastructure based wireless network through access points which are themselves interconnected through wired or wireless networks. A wireless ad hoc network may be better able to support mobile devices since these devices may still be able to communicate through other nodes even when they move outside the radio range.
of the fixed access points. In contrast, an infrastructure based wireless network can only support mobile devices that are within the direct radio range of at least one of the access points.

(a) Ad hoc Wireless Network

(b) Infrastructure Based Wireless Network

Figure 1.1. Comparison of Two Types of Wireless Networks
Chapter 1 Introduction

The capability of supporting infrastructure-less communication requires the wireless devices to be equipped with both wireless communications and networking capabilities. They can communicate directly, or via one or more intermediate devices or nodes. They are self-organizing and adaptive in nature and are able to detect the presence of other nodes and start communications thereafter. The wireless devices will have to take care of routing in both static as well as mobile situations, as there are no access points or base stations or fixed routers to assist in networking.

Routing protocols designed for wireless ad hoc networks are categorized into four groups [1] based on (a) routing information update mechanism, (b) use of temporal information for routing, (c) routing topology and (d) utilization of specific resources, as shown in Figure 1.2. In the first category, table-driven routing protocols require all nodes to maintain the routing information in their route tables by periodically exchanging routing information. On the other hand, on-demand routing protocols will obtain the necessary route as and when required using a connection establishment process. Hybrid routing protocols combine the best features of both table-driven and on-demand routing protocols. The second category of routing protocols uses past temporal information (like link status) or predicted future temporal information for route selection. In the third category, routing protocols may be based either on a flat topology using a flat addressing scheme similar to the one used in IEEE 802.3 LANs [26], or on a hierarchical routing topology using a logical hierarchy and an associated addressing scheme. The last category of routing protocols takes into account the utilization of specific resources like battery power resource or geographic information.
Figure 1.2. Categorization of Routing Protocols for Wireless Ad hoc Networks  
(extracted from [1])

Power conservation and efficiency issues are not unique to wireless ad-hoc networks but do have more serious repercussions here than in fixed networks. These issues may not arise in infrastructure based networks with nodes which have constant power available. Moreover, in the case of an infrastructure based wireless network, data forwarding is only done by the routers and the user nodes (mobile or otherwise) do not spend any of their own power in routing and forwarding packets for other nodes. In such networks, even if there are mobile devices running on limited battery power, these nodes do not incur power expenditure in carrying out data forwarding and routing for other nodes. In contrast, power expenditure in data forwarding and routing will be a serious issue in a wireless ad hoc network because of two conflicting requirements – i.e. that the network will not function if nodes do not route/forward packets to/from other nodes but that
forwarding packets for other nodes will drain the resources of a node without any direct benefits to the node itself.

Ongoing research activities in the areas of power conservation and efficiency have targeted different layers of the network protocol stack such as the physical, medium access control (MAC), network, transport, or application layers. However, most such research has been targeted at the MAC layer and the network layer. For the first part of the thesis, power efficiency at the network layer will be addressed, with the intention of developing an on-demand routing protocol that addresses the limited battery power concern of individual nodes.

In a wireless ad hoc network, when nodes form up to communicate in an ad hoc manner, a node will want to spend most, if not all, of its limited battery power resource for self transmissions. The individual goal of each node would be to get as much of its own communication through the network as possible. In cases where all nodes are selfish and do not cooperate to forward and route for each other, the network will be unable to make multi-hop data transmissions. Even if only a few nodes are willing to forward, multi-hop transmissions will be limited, and the system will unfairly use up the power resources of these cooperative nodes for forwarding and routing. There is a need to get all nodes to cooperate and participate in forwarding and routing, and also to ensure that the power resource of each node is used fairly for these networking activities. Various cooperation enforcement approaches have been proposed for nodes to cooperate in sharing their battery power for forwarding and routing. These may be classified as Reputation-based, Reward-based or Behavioral-based systems. Nodes in these systems are made to play a game of decision making (on whether to forward or not) that will
also affect their self transmissions. The second part of the thesis will focus on designing an on-demand routing protocol based on a proposed Reward-based system.

Selecting the shortest path may no longer be the sole objective of routing in a wireless ad hoc network. Considering the battery power constraint of each node in the case of a static network, route selection should take into account two power aware objectives, i.e. transmission power and/or remaining node battery capacity. Always selecting a route with the minimum total transmission power required will lead to efficient usage of the overall available energy resources in the network. On the other hand, always selecting a route that maximizes the minimum node battery power during route selection will prolong the network lifetime where network lifetime is defined as the time elapsed till the first node has depleted its battery power resource. In the case of a mobile network, an additional mobility aware objective, i.e. the established route lifetime that will be affected by node movement, needs to be considered. When a forwarding node moves out of range and breaks a link, the affected route lifetime would have expired and this may adversely affect ongoing data transmissions and the end-to-end delay performance of a connection. To obtain the transmission power and route lifetime information, each node has to know its location precisely, either from a Global Positioning System (GPS) or by using some other localization technique. Recent work [27] states that the requirement for each node to be equipped with GPS is quite realistic today because it is quite inexpensive and can provide reasonable precision. A number of authors have assumed the presence of GPS [27]-[34] for location information while others, such as [28], suggest using localization techniques to obtain location information in the absence of GPS. The third part of the thesis will focus on designing on-demand routing
protocols that address the various objectives stated for the static and mobile networks, by making use of the available location information.

1.2 Objectives and Contributions of the Thesis

The main objective of this thesis is to design and develop routing protocols for wireless ad hoc networks that addresses the three major concerns identified above, i.e. limited node battery capacity, cooperation between nodes in carrying out networking activities, and routing based on satisfying multiple objectives that include power and mobility awareness objectives. To address the first concern, power-aware node and route cost functions that are both dependent on the remaining battery power capacity of a node are proposed. The contribution of this thesis is to modify the existing Ad hoc On-Demand Distance Vector (AODV) [14], [35]-[36] routing protocol by incorporating these cost functions. The new protocol devised in this fashion has been referred to as Power Aware AODV (PAW-AODV). The simulation results show that PAW-AODV can carry more packets (i.e. higher data throughputs) by virtue of its power aware mode of operation while providing substantially similar delay performance compared to standard AODV protocols. PAW-AODV is also studied and compared with AODV in various mobile scenarios and in situations where different hop count limits are imposed in order to obtain a comprehensive idea of its overall performance.

To address the node cooperation issue, a set of nodes is considered to be playing a data forwarding game based on a Reward or Cost-Credit system. From this simple scenario, a forwarding rule is designed. The contribution here is to adapt this forwarding rule to a more realistic scenario and integrate this rule into a proposed On-Demand Cost-credit
routing (ODCCR) protocol for cooperative routing. ODCCR can be applied to power constrained ad hoc networks, which may be either static or mobile. This uses a route management module to take care of routes with forwarding nodes that have subsequently become unavailable due to deactivation (no more battery power left) or mobility. The implementation of a forwarding rule in ODCCR ensures that each node will be able to increase their self transmission in this cooperative and power constrained environment. Simulation results show that ODCCR performs very well in terms of network lifetime and throughput performances. The cost-credit system is further analyzed to formulate an optimization problem that aims to maximize the minimum battery power among all nodes when selecting routes for all the source nodes. Routes selected must enable source nodes to transmit as many data packets as possible. A centralized routing protocol called max-min Power Credit Routing or mmPCR is then proposed to handle this route selection task. mmPCR is evaluated against ODCCR for a comparative study of their respective strengths and weaknesses.

The final contribution of this thesis is to formulate various multi-objective routing problems and propose corresponding on-demand Location aided Multi-objective routing (LAMOR) protocols for practical static and mobile ad hoc networks. For static networks, three variations of a power aware multi-objective routing problem are formulated. These can be solved using a known optimization technique. The two main power aware objectives are maximizing the minimum node battery power and minimizing the total transmission power required to reach the destination. As it is not practical to employ an optimization technique to solve and select routes in large networks, three on-demand Location-aided Multi-objective Routing (Power Aware version) protocols or LAMOR-PA based on the proposed heuristics have been proposed. Each proposed protocol can
achieve results close to those obtained via optimization, and all three protocols perform better than other power aware routing protocols. With the presence of location information, a flood control mechanism to control the flooding of control traffic during route discovery is proposed. Since only location information is present, a mobility prediction module is proposed so as to obtain the speed and heading direction of each node. All three parameters are required to compute the link and route lifetimes. With route lifetime given, another three \textit{LAMOR-PMA} (Power and Mobility Aware version) protocols are proposed for mobile networks. Performance studies show that the \textit{LAMOR-PMA} protocols perform very well and the proposed flood control mechanism is found to be very effective in reducing flooding of control traffic. The final contribution here is essentially to make full use of the available location information to address the multiple concerns of a wireless ad hoc network, i.e. energy efficiency due to limited battery capacity, stable routing in the presence of mobility, and restricted flooding of control packets that include route request and error packets.

1.3 \hspace{0.5cm} \textbf{Organization of the Thesis}

This chapter gives an introduction to the wireless ad hoc network and identifies three major concerns pertaining to routing that have to be addressed, setting the direction and objectives for this thesis. In Chapter 2, on-demand routing will be explained using the examples of \textit{AODV} and \textit{Dynamic Source Routing (DSR)} [13], [35], [37]. The discussion on these two on-demand protocols will provide some basic knowledge pertaining to the various on-demand routing protocols that will be designed and developed to address the concerns mentioned. Related works, namely works that address power awareness,
cooperation of nodes in forwarding and routing and multi-objective routing will be discussed in Chapter 2 as well.

Chapter 3 presents a detailed description of the proposed Power Aware AODV (PAW-AODV) routing protocol. This chapter describes the major operations like Route Discovery, and Route and Cost Maintenance for the proposed protocol. The improvements that are proposed to augment standard AODV protocol are highlighted. Some simulated scenarios for testing both AODV and PAW-AODV are described. Results from the simulation runs are presented for discussion. The two protocols are compared to provide a perspective of their relative merits to show how they would function in power constrained scenarios.

In Chapter 4, a forwarding rule based on the proposed forwarding strategy and ODCCR are presented. This chapter also presents the ODCCR system’s performance in terms of network lifetime and throughput, and compares that to a typical AODV protocol and a Reward-based On-demand Routing (RBODR) protocol when used in a power constrained static ad hoc network. RBODR is an on-demand routing protocol based on a reward system but without any forwarding rule implemented. The performance studies also consider these protocols when operating in a mobile environment. An optimization problem that aims to maximize the minimum battery power among all nodes when selecting routes for all the source nodes in a cost-credit based system is presented next. The centralized routing mmPCR that solves the optimization problem and performs centralized routing is discussed. Its network lifetime and throughput performance results are discussed as well.
Chapter 1 Introduction

In Chapter 5, a general power aware multi-objective problem is formulated, followed by three variations of the power aware multi-objective problem. Next, a description of the proposed flood control mechanism and the three LAMOR-PA protocols corresponding to the three variations of the power aware multi-objective problem is given. The results of the performance studies conducted on LAMOR-PA in static networks are then discussed. The results of LAMOR-PA are also compared with their corresponding optimized results. The optimized results are obtained by solving the multi-objective problems using standard optimization techniques. The optimized results represent a baseline for comparison as they represent the best that can be achieved in principle under the given conditions. Considering the node mobility requirement, another three power and mobility aware multi-objective problems are formulated. A mobility prediction module is proposed so as to obtain the speed and heading direction of each node. Lastly, the three corresponding LAMOR-PMA protocols and their performance results from simulated mobile scenarios are discussed.

In the final chapter, conclusions regarding the research work reported in this thesis will be made, followed by some recommendations for future research.
Chapter 2 Literature Review

This chapter gives an introduction to on-demand routing for ad-hoc networks and reviews the relevant research that address the concerns and problems identified earlier in Section 1.1. Section 2.1 introduces on-demand routing using two commonly used on-demand routing protocols, AODV and DSR, as examples. A comparison between these two protocols is also given. Next, a review of some MAC layer and Network Layer Power Aware protocols that address battery power constraint and energy usage is given in Section 2.2. Three categories of Cooperation Enforcement Techniques are identified, namely Reputation-based, Reward-based and Behavioral-based systems. Research available in the literature on each category will be discussed in Section 2.3. Lastly, some proposed multi-objective routing protocols are reviewed and summarized in Section 2.4.

2.1 On-demand Routing

2.1.1 Ad hoc On-demand Distance Vector Routing

As its name implies, the Ad hoc On-demand Distance Vector (AODV) [14], [35]-[36] routing protocol discovers routes as and when needed, and maintains these routes as long as they are active. Figure 2.1 illustrates the AODV Route Discovery process. A source node that does not have a route to the destination will broadcast Route Request (RREQ) packets, as indicated by the continuous length arrows. The neighboring nodes will re-broadcast these RREQ packets if they, too, do not have a path to the destination node. Each neighboring node will record, in its routing table, the IP address of the immediate upstream node that forwards the RREQ packet to it. This is for the purpose of establishing a reverse route back to the source node later on. To prevent over-
flooding and looping of RREQ packets, if a node has received the same RREQ packet from the source node within a flooding period as defined in [36], it will not re-broadcast the RREQ packet further. Once the destination node receives the RREQ packet, it will reply with a Route Reply (RREP) packet. While this RREP packet travels back to the source node along the reverse route (as indicated by the dashed arrows in Figure 2.1), an intermediate node along this route will record, in its routing table, the IP address of the immediate downstream node that forwards the RREP packet to it. This establishes the forward route to the destination node for data transmission.

There is a timeout mechanism to determine how long a route can stay active. A lifetime value is associated with each route entry in the routing table. This lifetime value is updated whenever the route is used, but expires if the route has not been used within a given lifetime. AODV is loop-free as it uses sequence numbers to keep routes fresh in the routing tables. Basically, each node maintains its own sequence number, and increases the sequence number each time there is a change in the topology of its neighborhood. Only routes with the most recently updated sequence numbers are used.

AODV is able to provide unicast, multicast and broadcast communication ability. Since the route information for multicast communications can also be used for unicast communications, AODV can reduce overall control traffic and simplify coding. AODV assumes the wireless links to be symmetric in nature. Apart from this symmetric link requirement, AODV does not impose any other requirements on the wireless physical medium. AODV may be used in both wired and wireless networks, even though it was designed specifically for wireless networks.
Chapter 2 Literature Review

AODV makes use of routing tables to store routing information like destination entry and the associated next-hop node, destination sequence number and list of precursor nodes. The precursor nodes are nodes which are upstream with respect to the node that stores the table, and which use the latter node to route to the destination. If a downstream link breaks, this node will inform its precursor nodes so that they can take the necessary remedial actions, if any.

AODV includes an optimization technique to control the RREQ flooding that takes place during the route discovery process. It initially uses an expanding ring search to discover routes to an unknown destination. In the expanding ring search, increasingly larger neighborhoods are searched to find the destination. The search is controlled by the Time-To-Live (TTL) field in the Internet Protocol or IP header of the RREQ. If the route to a previously known destination is needed, the prior hop-wise distance is used to...
optimize the search. This enables the TTL value (used in the RREQ) to be computed dynamically by taking into consideration the temporal locality of routes.

*AODV* is an efficient routing protocol as it does not generate traffic unless necessary and removes unwanted and outdated information. With the use of a special route error (RERR) message, it is also able to remove invalid routes in a timely manner. Because of these features, *AODV* can respond to topological changes that affect active routes in a quick and timely manner. It can construct routes without having to generate huge control traffic and does not incur very high additional network overheads. As it only stores the next-hop information and a single route for each destination, the storage demands on each node is also reasonable. Moreover, it does not require additional overhead to be added to data packets as it does not perform source routing.

### 2.1.2 Dynamic Source Routing (DSR) Protocol

Dynamic Source Routing or *DSR* [13], [35], [37] protocol is another on-demand routing protocol. Like *AODV*, it also makes use of a Route Discovery process to find a route from the source to the destination, if the source does not already know a route to the destination. However, in *DSR*, a node may learn and cache multiple routes to any destination. This does not happen in the case of *AODV*. This strategy allows a quick response to routing changes caused by a change in the network topology, as a node can then select another route from the multiple routes stored in its cache whenever the route it was using earlier fails. Caching reduces the overhead incurred, as the node does not need to perform Route Discovery again each time a route fails. Figure 2.2 shows the Route Discovery process in *DSR*. As in the case of *AODV*, RREQ packets are broadcast
in search of required routes for data transmission, and RREP packets are returned to establish forward routes to destination nodes. RREQ packets from the same source within a discovery cycle are discarded if received more than once. Comparing with Figure 2.1, an obvious difference is reflected in the brackets in Figure 2.2. These show that intermediate node addresses are included in the RREQ as well, and that these are finally added into the node caches when the RREP is received. This way, apart from the route to the actual destination, the source will also learn routes to each intermediate node in the path followed to that destination. The intermediate nodes can also learn routes to every other node on the route.

The source broadcasts a RREQ to discover route

The node looks up its route cache to search for a route to destination
If not found, appends its address into the RREQ

Path traveled by RREQ

Path traveled by RREP

Figure 2.2. Route Discovery in DSR

The Route Maintenance process in DSR allows a node to either select another route from the cache (if one exists), or invoke a Route Discovery process, whenever the network topology changes to cause a break in the existing route during data transmission. This process may also be invoked on-demand, e.g. when a node is actually
sending packets and then discovers a route or link failure. Thus, all processes in DSR are invoked on-demand, without the need to use periodic routing advertisement, link status sensing or neighbor detection packets (unlike the approach in AODV). The purely on-demand behavior and lack of periodic activity is needed to reduce the DSR overhead. This will be especially true when the nodes are approximately stationary with respect to each other and all required routes have been discovered. In contrast, the DSR overhead increases proportionally when nodes move or when the communication pattern changes.

It may be noted that unlike AODV, DSR supports unidirectional links and asymmetric routes as well. It also supports internetworking between different types of wireless networks, allowing a source route to be composed of hops over a combination of different network types that may be available.

2.1.3 Protocol Comparison

AODV and DSR are both on-demand routing protocols [1], [35]. This means that routes are discovered only when data packets need to be transmitted but there are no valid routes to the destinations at that particular moment. However, there are some important differences between AODV and DSR.

An obvious difference is that route information is stored in route tables of the intermediate nodes for AODV, while for DSR, route information is stored in route caches of the intermediate nodes. This implies that the storage requirements for DSR would be higher as route caches for DSR may contain multiple paths per destination.
Chapter 2 Literature Review

while AODV route tables store only one single route entry per destination. DSR is able to learn many paths to each destination as it replies to all requests reaching a destination from a single request cycle. In this way, the source learns many alternate routes to the destination, which will be useful if the primary (shortest) route fails. Having access to many alternate routes reduces route discovery floods, which is often seen as a performance bottleneck. However, there may also be a possibility of a route reply flood. In AODV, on the other hand, the destination replies only once to the request (i.e. to the one arriving first) and ignores the rest. Therefore, the routing table maintains only one entry per destination.

For DSR, using a single request-reply cycle, the source can learn routes to each intermediate node on the route in addition to the intended destination. Each intermediate node can also learn routes to every other node on the route. Promiscuous listening of data packet transmissions gives DSR access to a significant amount of routing information. In particular, it can learn routes to every node on the source route of that data packet. On the other hand, without the use of source routing and promiscuous listening, AODV cannot gather as much routing information as DSR. Route learning is limited only to the source of any routing packets being forwarded. This usually causes AODV to flood more often to discover routes, thereby incurring higher network overheads.

DSR does not contain any explicit mechanism to expire stale routes in the cache, or to prefer “fresher” routes when faced with multiple choices. Stale routes, if used, may actually start polluting other caches. Some stale entries are indeed deleted by route error packets. However, because of promiscuous listening and node mobility, it is possible
Chapter 2 Literature Review

that more caches are polluted by stale entries than are removed by error packets. In contrast, AODV has a much more conservative approach than DSR. When faced with two choices for routes, the fresher route (based on destination sequence numbers) is always chosen. Moreover, if a routing table entry is not used recently, the entry is expired. However, it is also possible that valid routes get expired this way if the routes remain unused beyond their expiry times. Determination of a suitable expiry time is difficult, because the sending rates for different sources, as well as node mobility, may not only differ widely but may also change dynamically.

The route deletion activity using Route Error or RERR packet is conservative in AODV. By way of a predecessor list, the error packets reach all nodes using a failed link on its route to any destination. In DSR, however, a route error simply back-tracks the data packet that meets a failed link. Nodes that are not on the upstream route of this data packet but use the failed link are not notified promptly.

2.1.4 Performance Comparison

Some researchers have attempted to compare the performance of AODV and DSR routing protocols [38]-[39] using experiments and simulation. Some of these simulation results (from [38]) are presented next to compare the relative merits of the aggressive use of source routing and caching in DSR with more conservative routing table and sequence-number-driven approach in AODV.

Figure 2.3 shows plots of packet delivery fraction and average delay against pause time, for varying number of sources [38]. Packet delivery fraction is the ratio of the data
packets delivered to the destinations to those generated by Constant Bit Rate (CBR) sources. Average delay is the end-to-end delay of delivering data packets, and this includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times. There are a total of 100 nodes in the test environment of [38] with each source transmitting at 4 packets/sec. The 40-sources scenario is slightly different with each source transmitting at 3 packets/sec. All data packets are of the same size of 512 bytes. The pause time indicates how mobile the nodes will be, with 0s pause time indicating the nodes never remain stationary at any location and 500s pause time indicating that a node will remain stationary at its current location for a period of 500s before moving off to another location.

From Figure 2.3, it can be seen that the performance of AODV and DSR are similar when there are only a few sources. When the number of sources increases, AODV is significantly superior to DSR both in terms of packet delivery ratios and average delays. This implies that as a protocol, AODV is more scalable than DSR which is a conclusion that has been generally reported in the research literature. Figure 2.3 also shows that AODV performs better than DSR when node mobility is high, i.e. the pause times are low. Figure 2.4 shows another set of results from [38] for simulations done with zero pause time (i.e. highly mobile nodes). In these scenarios, the total offered load is increased and the throughput, route load and average delay results are recorded. The throughput is the combined received throughput at the destinations while the offered load is the combined sending rates of all data sources. The route load is the load overhead contributed by the RREQ, RREP and RERR packets and is the number of these routing control packets transmitted per second. Even though AODV generates
more route load than DSR, it nevertheless performs better by providing higher throughputs and lower delays. In line with the better scalability of AODV mentioned earlier, this superiority of AODV over DSR becomes more evident with an increasing number of sources.

Figure 2.3. Packet Delivery Fraction and Average Data Packet Delays for a 100-node Network with Different Numbers of Sources (extracted from [38])
In summary, based on the results of the simulation runs, it is found that considering application-oriented metrics such as delay and throughput, DSR outperforms AODV in less “stressful” situations, such as networks with smaller number of nodes and lower load and/or less node mobility. However, AODV outperforms DSR in more stressful situations, with widening performance gaps with increasing stress (i.e. with more load and/or higher mobility). However, DSR consistently generates less routing load than AODV [38]. The poor delay and throughput performances of DSR are mainly attributed to the aggressive use of caching, and lack of any mechanism to expire stale routes or determine the freshness of routes when multiple choices are available. However, it is the aggressive use of caching that also actually causes DSR to perform well at low loads and also allows it to keep its routing load down. Overall, AODV is a more scalable protocol than DSR. Table 2.1 summarizes the differences between AODV and DSR.

Figure 2.4. Performance with Increasing Offered Load for 100 nodes with 10 and 40 Sources (extracted from [38])
### Table 2.1. Differences between AODV and DSR

<table>
<thead>
<tr>
<th>Features</th>
<th>AODV</th>
<th>DSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage requirement at nodes</td>
<td>Less since route tables keep a single route entry per destination</td>
<td>More since route caches keep multiple route entries per destination</td>
</tr>
<tr>
<td>Number of discovered routes recorded per destination</td>
<td>One</td>
<td>More than one, if found</td>
</tr>
<tr>
<td>Amount of route information learned per request-reply cycle</td>
<td>Less as source only learns route to destination</td>
<td>More as source learns route to destination as well as intermediate nodes</td>
</tr>
<tr>
<td>Request flooding frequency</td>
<td>More due to limited route information learnt per request-reply cycle</td>
<td>Less to the additional information learnt per request-reply cycle</td>
</tr>
<tr>
<td>Control traffic per request-reply cycle</td>
<td>Less as less route discovery traffic is generated to establish a single path per destination in the route table</td>
<td>More due to more route discovery communications required to establish multiple routes per destination in route cache</td>
</tr>
<tr>
<td>Freshness of routes recorded and used</td>
<td>Presence of a timeout mechanism to keep recorded routes updated, and an algorithm to always choose the most updated route</td>
<td>No such explicit mechanism or algorithm</td>
</tr>
<tr>
<td>Effect of route error notification mechanism</td>
<td>Effective since all nodes affected by the broken link will be informed</td>
<td>Less effective since only nodes on the upstream route of this data packet that has encountered the broken link will be informed</td>
</tr>
<tr>
<td>Performance comparison based on throughput and delay</td>
<td>Performs well in networks with high traffic load / huge number of nodes / high node mobility</td>
<td>Performs well in networks with less traffic load / reasonable number of nodes / low node mobility</td>
</tr>
</tbody>
</table>

### 2.2 Power Aware Protocols

#### 2.2.1 MAC Layer Power Aware Protocols

One can consider two categories of MAC layer protocols [40], i.e. single channel and multiple channel protocols. Single channel algorithms require all nodes to transmit on
the same channel, and the access control approach is based on random access, coordinated access or scheduling access. MAC protocols using multiple channels require nodes to transmit on separate channels. Basically, a channel is divided into multiple channels using approaches such as code division multiple access (CDMA) [41], time division multiple access (TDMA) [41], frequency division multiple access (FDMA) [41] or orthogonal frequency division multiple access (OFDM) [41]. The algorithms then assign the available channels to the nodes such that no collision (or very few collisions) would occur. In effect, this reduces retransmissions and will lead to greater power conservation. As a side benefit, these algorithms also tend to maximize the number of simultaneous transmissions. This tends to reduce delays and leads to better overall usage of the system’s resources.

Among the control methods mentioned earlier, coordinated access is currently the main access control method used. For example, this is the primary access control protocol in the IEEE 802.11 specifications [4]. As a result, a number of MAC layer power aware protocols have been proposed based on coordinated access control. Power Aware Multiple Access Scheme (PAMAS) [42], Power Control Multiple Access (PCMA) [43] and an approach using battery power state for MAC layer operation [44] are relevant examples of such schemes.

*PAMAS* uses different channels for data and control packets so as to reduce collisions and retransmissions, thereby increasing power savings. A sleep-and-wake scheme is also proposed, to further reduce the power consumed in the idle state as this saving can be quite substantial. A node that wishes to transmit but overhears a busy tone from its receiver will go to ‘sleep’ so as not to waste any energy in the ‘idle’ state. If the node
Chapter 2 Literature Review

go to sleep when a transmission just begins in its neighborhood, it will know from the length of the transmission when to wake up. If the node decides to sleep in the middle of the transmission, it will send a probe signaling packet for the purpose of checking when the transmission will finish, so that it can then decide when to wake up. The power savings resulting from this may be significant in dense networks.

The basic idea behind PCMA is to tag the power information in the request-to-send (RTS) and clear-to-send (CTS) packets used in IEEE 802.11 (or other similar protocols). This is done to allow nodes to adjust their own transmission power level so that they can transmit without interfering with neighboring transmissions. This would reduce collisions thereby increasing the number of simultaneous transmissions. It will also lead to more efficient utilization of the node’s own power resources (i.e. its battery). Here, the receiver first calculates the optimum power that the sender should use for transmission. This is done from the transmitted power and interface noise level information conveyed by the Request-Power-to-Send or RPTS (equivalent to RTS) packet, and also from the received power level. Then this optimum transmission power information is conveyed back to the sender by the Acceptable-Power-to-Send or APTS (equivalent to CTS) packet. Every receiver is supposed to maintain its signal-to-noise ratio above a threshold value in order to receive packets without errors. To achieve this, the receiver transmits a busy tone on a separate channel with a power level inversely proportional to its noise tolerance. Any neighboring node that intends to transmit but receives this busy tone will decrease its transmission power level in proportion with the received power level of the busy tone. It is noted that this proposal may also be effective in tackling the common exposed terminal problem in wireless networks.
Chapter 2 Literature Review

The research reported in [44] designs a distributed Battery Aware Medium Access Control (BAMAC(k)) protocol. This takes advantage of the chemical properties of batteries, in order to provide fair scheduling along with higher network and node lifetimes through uniform discharge of batteries. Basically, the Battery Aware MAC (BAMAC(k)) protocol tries to increase the lifetime of the nodes by exploiting the recovery capacity effect of the battery. If the battery remains idle for a specified time interval, it becomes possible to extend the lifetime of the battery due to the recovery capacity effect. This behavior is illustrated in (2.1), which shows that this effect will be higher when the battery has higher remaining capacity and decreases with a decrease in the remaining battery capacity.

\[
R_{N_i,T_i} = e^{-g(N-N_i)-\Phi(T_i)} \quad \text{if } 1 \leq N_i \leq N \\
= 0 \quad \text{otherwise}
\]  

(2.1)

where \( R_{N_i,T_i} \) is the probability that the battery recovers one unit of the charge if the node remains in the receive state and the neighbor nodes transmit but this node does not receive any packets,

- \( N \) is the nominal battery capacity,
- \( N_i \) is the remaining battery at time \( i \),
- \( T_i \) is the (transmission) state at time \( i \),
- \( g \) is a constant,
- \( \Phi(T_i) \) is a piecewise constant function of number of charge units delivered which are specific to the battery's chemical properties.

The BAMAC(k) protocol tries to provide enough idle time for the nodes of an ad hoc wireless network by scheduling the nodes in an appropriate manner. It tries to provide uniform discharge of the batteries of the nodes that contend for the common channel. This can be done by using a round-robin scheduling (or fair-share scheduling) between
these nodes. To implement a round-robin scheduling of the nodes in a distributed manner, each node maintains a battery table which contains information about the remaining battery charge of each of its two-hop neighbor nodes. The entries in the table are arranged in non-increasing order of the remaining battery charges. The RTS, CTS, Data, and ACK packets carry the following information - (a) remaining theoretical (in terms of remaining battery voltage) and (b) nominal capacities of the battery and (c) the time of last usage of the battery (the time at which the battery underwent its last discharge) for the node that originated the packet. A node, on listening to these packets, makes a corresponding entry in its battery table. The objective of the back-off mechanism used in BAMAC protocol is to provide a near round-robin scheduling of the nodes. The back-off period is given by:

$$back-off = \text{Uniform}[0, (2^x \times CW_{min}) - 1] \times rank \times (T_{SIFS} + T_{DIFS} + T_i)$$  \hspace{1cm} (2.2)$$

where $CW_{min}$ is the minimum size of the contention window,

$x$ is the number of transmission attempts made so far for a packet,

$rank$ is the position of that entry in the battery table of the node which is arranged based on the following rule: “The battery table is arranged in descending order of its theoretical capacity of the nodes. Any tie, that arises, is broken by choosing the one with higher nominal capacity and then by choosing the one with least value for the time of last usage. Further ties are broken randomly”,

$T_{SIFS}$ represents the SIFS (Short inter-frame spacing) duration,

$T_{DIFS}$ represents the DIFS (DCF inter-frame spacing) duration,

$T_i$ is the longest possible time required to transmit a packet successfully, including the RTS-CTS-Data-ACK handshake.
When this back-off scheme is followed, nodes with lesser rank values back-off for smaller time durations are comparable to those with higher rank values. \( \text{Uniform}[0, (2^i \times CW_{\text{min}}) - 1] \) in (2.2) returns a random number distributed uniformly in the range 0 and \([ (2^i \times CW_{\text{min}}) - 1 ]\). This implies that the nodes are scheduled based on their remaining battery capacities. The higher the remaining battery capacity, the lower would be the back-off period. This ensures a near round-robin scheduling of the nodes. Hence, a uniform rate of battery discharge is guaranteed across all the nodes.

### 2.2.2 Network Layer Power Aware Protocols

#### Routing based on Power Aware Cost Function

Various research works have suggested the use of power aware approaches to route traffic. A link energy consumption function, based on link error probability and power consumption, has been proposed in [45]. This proposes using a link cost metric as in (2.3).

\[
C_{i,j} = \frac{E_{i,j}}{1 - p_{i,j}}
\]

(2.3)

Here, for link \( ij \), \( C_{i,j} \) is the link cost, \( E_{i,j} \) is energy or power required for transmission and \( p_{i,j} \) is link error rate. Note that this combines the link’s error rate and power consumption in one cost metric for the link and is a good measure of how much energy will be consumed in a single link, based not only on the transmission power required, but also on the extra power required for retransmission due to the link error rate. However, this may lead to over-usage of certain nodes, draining their power quickly and eventually disconnecting them from the network faster than expected.
An alternative routing protocol proposed in [46] is one where packets from a source are preferentially routed through those neighboring nodes which have the least amount of data waiting to be transmitted. This will reduce the variance in the power level of the nodes by trying to balance the work load between them. Here, the node’s cost function is tailored to represent its battery’s remaining life time. In this case, if $c_i(t)$ is the battery capacity of node $n_i$ at time $t$ then the node’s cost function may be given as

$$f_i = 1/c_i(t) \quad (2.4)$$

This implies that a node with lower (remaining) battery capacity will result in a higher node cost. Here, the route cost is taken as the maximum node cost of all the intermediate nodes that lie along the route. This means that the route cost function for a route $r_j$ is given by

$$R(r_j) = \max_{n_i \in r_j} f_i \quad (2.5)$$

Here, the routing strategy chooses the route with the minimum route cost as defined in (2.5). This approach will then choose a route whose weakest node (i.e. the one with the lowest remaining battery capacity) has the highest residual power among all the potential routes that are considered. This reduces the variance in power level of the nodes in the system but has the disadvantage that it does not consider the total power consumed in the selected path. Therefore, the total power consumption may not be the lowest along the selected path.

Another improvement that can be made to the above algorithm is given in [47]. Here, all routes whose nodes are above some threshold (remaining) power level are first selected. Then these selected nodes are used to find the route which will consume the least overall power. If all the routes have nodes with remaining power below the threshold value, then the min-max battery cost routing of [46] is applied. Apart from being more
complex than [46], deciding the right threshold value becomes crucially important as the protocol’s performance depends strongly on this choice.

Each node can keep track of its energy consumption and battery power drain rate value $DR$ by averaging the amount of energy consumption and estimating the energy dissipation per second during the past given interval [48] using (2.6)

$$DR_i = \alpha . DR_{old} + (1-\alpha).DR_{sample}$$  \hspace{1cm} (2.6)

Here, the drain rate $DR_i$ of node $n_i$ is derived from the drain rate $DR_{old}$ obtained from the past interval and the sampled drain rate $DR_{sample}$ from the current interval. $\alpha$ is set to 0.3 [48] to give higher priority to the current sampled drain rate so as to better reflect the current condition of energy expenditure of nodes. In this case, (2.7) gives an estimate of when the remaining battery of a node $n_i$ will be exhausted, whereby $RBP_i$ is the residual battery power of node $n_i$. The strategy then is to compute the maximum lifetime of a given path $r_p$, as reflected by the minimum value of $C_i$ over the path as given in (2.8). Then the route $r_M$, contained in the set of all possible routes $r$, between the source and the destination node, that presents the highest maximum lifetime value is selected as reflected in (2.9). This strategy can avoid overloading a few nodes because their remaining power levels are high at some point in time. However, this cannot guarantee minimum power consumption.

$$C_i = \frac{RBP_i}{DR_i}$$  \hspace{1cm} (2.7)

$$L_p = \min_{r \in r_p} C_i$$  \hspace{1cm} (2.8)

$$r_M = \max_{r \in r} L_i$$  \hspace{1cm} (2.9)
Power-aware Source Routing (PSR)

A Power-aware Source Routing (PSR) algorithm that aims to extend the lifetime of a mobile ad hoc network or MANET is proposed in [49]. The lifetime of the network is defined as the time taken for a certain percentage of nodes in the network to “die” due to complete energy depletion. This objective is of primary importance since the power exhaustion of some of the nodes can cause communication breakdown even if the source and destination nodes still have ample battery power. PSR attempts to find a route $\pi$ such that the following route cost function is minimized.

$$C(\pi, t) = \sum_{i \in \pi} C_i(t)$$  \hspace{1cm} (2.10)

where

$$C_i(t) = \rho_i \left( \frac{F_i}{R_i(t)} \right)^{\alpha}$$ \hspace{1cm} (2.11)

Here, for node $i$, $\rho_i$ is the transmit power, $F_i$ is the full-charge battery capacity of the node, $R_i(t)$ is the remaining battery capacity of the node at time $t$ and $\alpha$ is a positive weighting factor.

PSR is actually an enhancement of Dynamic Source Routing (DSR). Like DSR, it discovers routes on demand and maintains routes in a similar fashion as DSR. In PSR, all nodes except the destination calculate their link cost, and add it to the route cost in the header of the RREQ packet. When an intermediate node receives a RREQ packet, it starts a timer $T_r$ and keeps the cost in the header of that packet as Min-Cost. If additional RREQ packets arrive with same destination and sequence number, the cost of the newly arrived RREQ packet is compared to the Min-Cost value. If the new packet has a lower cost, Min-Cost is changed to this new value and the new RREQ packet is forwarded. Otherwise, the new RREQ packet is dropped.
The destination will wait for a threshold time of $T_r$ seconds after the first RREQ packet arrives from a given source. During this time, the destination examines the cost of the route of every RREQ packet that may arrive from the same source. When the threshold time $T_r$ expires, the destination node selects the route with the minimum cost received during this time and replies back to the source. Any other RREQ packets received from the same source subsequently will be dropped. The reply sent back also contains the cost of the selected path appended to it. Every node that hears this route reply adds this route along with its cost to its route cache table. It has been claimed in [49] that this PSR scheme provides significant savings in cost even though it does increase the average latency of the data transfer (i.e. the mean delay) to some extent.

From the point of view of the route maintenance required in this scheme, it is noted that apart from the usual route maintenance activity due to node mobility, PSR will also need to perform route maintenance due to energy depletion. For this, it adopts the approach that starting from the time of route discovery, each intermediate node in a path monitors the decrease in its remaining energy level (and hence the resultant increase in its link cost) as a result of forwarding packets along this route. When this link cost increase goes beyond a pre-decided threshold level, the node sends a route error message back to the source just as if the route has been rendered invalid. This route error message forces the source to discard the current route and forces it to initiate the route discovery process once again.

Figure 2.5 shows the result obtained from the simulation runs done in [49]. Compared to DSR, PSR results in nodes “dying” at a later time, thereby extending the operational lifetime of the MANET. Since throughput and delay performances were not studied in
[49] it is not known how PSR’s increased control packet overhead (i.e. transmitted for route maintenance) and its higher latency in discovering routes affect these measures.

![Figure 2.5. Number of Dead Nodes in DSR and PSR versus Time](extracted from [49])

**Routing based on Topology Control**

There are a number of ways to achieve power conservation via topology control. An analytical expression can be derived to determine the transmission range $r$ that creates, for a given node density, an almost surely $k$-connected network [50]. If the maximum value of $r$ is given, then the number of nodes required to cover a certain area with a $k$-connected network can be determined. Alternatively, the smallest sub-graph of the given graph that contains the shortest path between all pairs of nodes can be used as the minimum power topology as in [51]. A topology problem can also be specified by \{M, P, O\}, where M $\in$ \{DIR, UNDIR\} represents the graph model (directed or undirected), P is the desired graph property and O is the minimization objective [52]. In Reference [52], an approximation algorithm is presented for the \{UNDIR, 2-NODE CONNECTED, TOTALP\} problem. This problem seeks to assign transmission powers to the nodes so that the resulting undirected (UNDIR) graph has a node connectivity of at least 2 (2-NODE CONNECTED) and the sum of the transmission powers (TOTALP)
assigned to all nodes is minimized. The approximation algorithm was analyzed using some properties of critically 2-node connected graphs [53]-[55]. With minor modifications, the authors also obtain an approximate algorithm for producing 2-edge connected graphs.

Another method proposed in [56] is to select a node as the information sink for all nodes in the network. A power conservation algorithm is then used to produce a minimum power topology between one node and all the other nodes. Though such topology can be obtained for each node, [56] does not consider obtaining a globally optimal topology from the individual topologies calculated for all the nodes. In this method, each node first searches for its possible set of neighbors. This algorithm selects the immediate neighbors based on the fact that routing through these immediate neighbors result in lower transmission power. Next, each node will broadcast its power consumption cost defined as $\text{cost}(i) = \min_{n \in N(i)} C_{i,n}$ to its neighbors and the distributed Bellman-Ford shortest path algorithm is then applied to select the minimum cost neighbor, with the power consumption cost as the cost metric. Let $n \in N(i)$ where $N(i)$ is the set of immediate neighbors of node $i$ found earlier, so when node $i$ receives the broadcast cost of node $n$, it computes the following -

$$C_{i,n} = \text{cost}(n) + P_{\text{transmit}}(i,n) + P_{\text{receiver}}(n)$$  \hspace{1cm} (2.12)

Here $P_{\text{transmit}}(i,n)$ is the power required to transmit from node $i$ to node $n$

$P_{\text{receiver}}(n)$ is the additional receiver power that node $i$’s connection to node $n$ would induce at $n$
Chapter 2 Literature Review

Then node $i$ will compute its cost as indicated by $\text{cost}(i) = \min_{n \in N(i)} C_{i,n}$ and picks the link corresponding to the minimum cost neighbor. The algorithm is repeated until it converges. It is noted in [56] that at the very beginning of the algorithm, an arbitrary cost may be assigned to node $i$ but that the costs will eventually converge.

Reference [57] has the same objective as [56], which is to design a location-based, distributed topology-control algorithm that increases network lifetime. The difference lies in the approach used. In [57], there are two phases of operation. In the first phase, starting with a small radius, each node $i$ broadcasts a neighbor-discovery message. Each receiving node returns an acknowledgement pertaining to this broadcast message. Node $i$ then records all acknowledgments and the directions they came from. It can then determine whether there is at least one neighbor in every cone of $\alpha$ degrees, centered on node $i$. In this phase, node $i$ continues the neighbor discovering process by increasing its transmission radius (operational transmission power) until either the above condition is met or the maximum transmission power is reached. In the second phase, the algorithm performs a redundant edge removal process without affecting the connectivity. This phase aims to reduce the node degrees, which helps in reducing interference and enhancing throughput. Redundant edge removal is carried out without deteriorating the minimum power routes of the network.

Routing based on Clustering

The principle behind using clustering for power conservation is to utilize a central coordinator to achieve the desired objective. An algorithm for the connecting dominating set problem can be used to select a coordinator in a cluster [58]. A
dominating set $S$ of a graph $G$ is a set of vertices for which each vertex of $G$ is either in $S$ or is incident to a vertex in $S$. A connected dominating set is a dominating set which forms an induced connected sub-graph of $G$. This coordinator can route packets for the group members, schedule sleep/wakeup activities for them and buffer the packets meant for its group members [i.e. as destination] who are currently sleeping. A disadvantage of this approach would be a possibly heavy drain on the power resources of the coordinator.

A scheduling scheme may be deployed to schedule the nodes so that a node which wants to transmit is allowed to do so for as long as possible before its state is changed from *transmitting* to *receiving*. This is proposed in [59] under the consideration that the price of changing state is high and that, therefore, nodes may wish to transmit for as long as possible before changing to the receiving state. The suggestion here is that the cluster coordinator may be used to perform such scheduling.

A dual clustering mechanism has also been proposed in [60]. In single clustering, in order to form a cluster, the master node pages at its maximum power level on a common control channel. Slaves receiving this page acknowledge with the received power level of the page. The master then allocates the channel it has to the slave on the basis of the received power level. This keeps the cluster small and any slaves not getting connections are not part of the network. In double clustering, the first phase is similar to the single clustering action. Once the master has assigned channels to slave nodes, if it has any channels left, it will re-page. The slave nodes who are rejected in the first round can re-acknowledge this time round to be re-selected again. The disadvantage is that this
is proposed for a static network and has not been examined in the mobile context. Moreover, the channel division may not be bandwidth efficient.

**Routing based on Coordinated Power Conservation**

Coordinated Power Conservation (CPC) [61] is a cluster-based approach which saves energy by turning off some nodes. It uses a *virtual dynamic backbone protocol* (VDBP) to construct a backbone such that it has the least number of nodes. VDBP comprises backbone selection, connection and maintenance processes. After backbone nodes are selected, they will be involved in power coordination. They will periodically check whether there are suspended requests from member client nodes and then decide which nodes should go to sleep and for how long. This is done based on the dynamics of the network, the number of active nodes per sleeping node, and the remaining battery power level of the client nodes. The backbone node then selects more stable nodes as candidate sleeping nodes. Figure 2.6 shows an operational cycle of a non backbone node. The obvious disadvantage is that the battery capacity of the backbone nodes will run out quickly compared to the other nodes of the network. One way to get around this is to alternate backbone node roles among all the nodes of the network. While this is a feasible idea to tackle the problem, it will impose additional overheads and complexity on the system.

A similar approach is proposed in the Geographical Adaptive Fidelity (GAF) scheme of [62] which uses geographical location information to divide the region into grids. Nodes within a grid can switch between sleeping and listening, as long as there is at least one node in the grid which is awake to route packets. A similar scheme, SPAN [63] rotates
the coordinators. It attempts to minimize the number of nodes elected as coordinators, so that the network does not suffer too much from capacity loss or higher latency.

![Diagram of CPC Non-Backbone Node](image)

Figure 2.6. Operational cycle of a CPC Non-Backbone Node

### 2.3 Cooperation Enforcement Techniques

Cooperation is important in wireless ad-hoc networks as these networks do not have specific nodes identified as routers. Intermediate nodes need to take up this role as and when required to ensure that a packet reaches its intended destination. It is important to note that all such intermediate nodes may also typically be source nodes themselves. Therefore, they must use their own power to transmit their own packets as well. If they wish to follow a selfish strategy to conserve their own power (to send only their own packets) then their obvious approach will be to report a high cost for forwarding packets from other nodes (even if they have sufficient power) or they may simply refuse to forward packets for others. Cooperation is not an issue in an ad hoc network set up for military or emergency purpose since all nodes have a common objective, i.e. to convey critical information in the network. This is not the case in a commercial or public network where each node has its own message to get across the network. Selfishness in such a situation is understandable as the node may wish to maximize its own benefits.
without regard to the benefits of the other nodes or of the overall system. However, this is counterproductive because the network will either not function properly if a significant fraction of its nodes decide to follow such a selfish strategy or will display poor performance with high delays and low throughputs. Thus, there is a need to implement a technique to encourage cooperation.

Consider a game [64] consisting of a set of strategies available to each player. In the game play, each player can choose a particular strategy, depending on what the other players have chosen, so as to maximize its payout function. (In this context, Nash Equilibrium is achieved when each player is playing a strategy that results in maximum payout, in response to the optimum strategies played by all other players). In different cooperation enforcement approaches, the nodes are the ‘players’ in the game and each node’s ‘strategy’ is either to forward or not to forward. Based on the strategies played, the ‘payout’ of each node will affect the amount of self transmission. The difference between these approaches lies in the way the ‘strategy’ used affects the ‘payout’. The three approaches to be discussed next are Reputation-based systems, Behavioral-based systems and Reward-based systems.

2.3.1 Reputation-based Systems

In Reputation-based systems, the forwarding behavior of nodes will be tracked and the reputation rating of a node will fall when it does not forward for others. If the reputation rating of the node falls below a threshold, other nodes will not forward multi-hop self-generated packets for this misbehaving node. CONFIDANT (Cooperation of Nodes: Fairness in Dynamic Ad-hoc Networks) [65] consists of four modules in each node:
Chapter 2 Literature Review

Monitor, Trust Manager, Reputation System and Path Manager. Figure 2.7 shows the trust architecture designed for CONFIDANT. Surrounding nodes will monitor and disseminate ALARM messages when a node is found to misbehave by not forwarding. Once the Trust Manager in a node finds the ALARM messages to be truthful, the reputation rating of the misbehaving node will be changed by the Reputation System. When the reputation rating of the misbehaving node falls below a threshold value, the Path Manager in the surrounding nodes will remove paths containing the misbehaving node and deny forwarding service to that node. The dissemination of reputation messages will incur extra network overhead and can be a source of denial of service attacks since the messages can be manipulated to deny forwarding service even to an obedient node.

Figure 2.7. Trust Architecture for CONFIDANT

CORE (A Collaborative Reputation mechanism to enforce node cooperation in mobile ad hoc networks) [66], [67] works in the similar fashion as CONFIDANT. The basic concept as described [66] is: each node will employ a Watchdog module to monitor the
behavior of its neighbors with respect to a requested function and collects observations about the execution of that function. If the observed result and the expected result coincide, the observation will take a positive value. Otherwise it will take a negative value. Based on the collected observations, each node computes a reputation value for every neighbor. When used together with DSR, node misbehavior can be detected during the route request and route reply phase. A game theoretical analysis [67] shows that if at least half of the total nodes cooperate, the rest of the nodes also have to cooperate in order to use the network.

PARS (Power Aware Reputation System) [68] can be built on top of CONFIDANT or CORE. It is an add-on system that prevents a node from falsely announcing its low energy level to avoid forwarding through itself in the case of power aware routing. It has a Detection module and an associated set of monitoring rules that monitor the energy reported in the routing packets. If a node is found to be providing false energy information, the Jury module in each neighboring nodes will be executed. Upon reaching an agreement, they will start to isolate the convicted node quickly and simultaneously for a period of time. PARS will monitor the energy-related reputation while utilizing CONFIDANT or CORE to monitor forwarding-related reputation, so when one of the reputation rating falls below a threshold value, the node concerned will be isolated and will be denied any forwarding service from its neighboring nodes.

### 2.3.2 Behavioral-based Systems

Behavioral-based systems play a ‘tit-for-tat’ game, whereby if a node does not forward for others, other nodes will behave similarly in time to come, by not forwarding for this
Chapter 2 Literature Review

node. For GTFT or Generous Tit-for-Tat algorithm [69], [70] proposed, the authors recognize that different nodes may have different battery power or energy constraints. The main aim is to balance the use of the limited power resource for self transmission and forwarding (relaying), by behaving in a manner similar to others. All nodes are classified under a number of energy classes, and a session of type $j$ is one with a connection (or route) with at least one node belonging to class $j$ (with the least power).

The GTFT algorithm states that a relay request for a type $j$ session is to be rejected if a relay node finds that it has successfully relayed too many type $j$ connections as compared to a computed value, or to the number of type $j$ connections initiated by it and successfully relayed by others. Using a game theory approach, it is shown that if all nodes play GTFT, then all nodes can be in Nash Equilibrium, achieving optimal throughput. This work only focuses on developing a mathematical framework for analyzing node cooperation and strategies for achieving optimal gain. Details on how to develop an algorithm to track system information like number of successful transmissions and relayed sessions, as required for the implementation of GTFT, are left as future research work.

A simple TFT (Tit-for-Tat) strategy is given in [71] as a way to encourage cooperation instead of using rewards or credits. The emphasis in this work is on the game theoretical analysis of the different simple behavioral strategies. The authors of [71] find that all nodes always defecting are in Nash Equilibrium, but this is unacceptable in a multi-hop ad hoc network. They also show that (i) if each node depends on other nodes for forwarding, and these other nodes depend on the node as well, and (ii) if the maximum forwarding cost for each node on every route where it is a forwarder is smaller than its possible future benefit averaged over all the routes (where it is a forwarder), then all
nodes playing *TFT* will result in a Nash Equilibrium. In reality, the above two conditions may not always exist for all nodes. Thus, some nodes may resort to the ‘always defect’ strategy and these nodes would need an incentive (probably rewards) to cooperate.

### 2.3.3 Reward-based Systems

*Reward-based* systems reward (with credits) a node that forwards for other nodes, so it can use the credits to pay other nodes for its multi-hop self transmission. One implementation is the use of a *nuglet* counter [72]-[73] maintained at each node, where *nuglet* is a virtual currency. When a node needs to transmit a self-generated data packet, its *nuglet* counter will be decremented. But the counter can be incremented again when this node forwards data packets for other nodes. When the *nuglet* counter goes to zero, which means a node has transmitted more self-generated packets than forwarded packets, it will not be allowed to transmit any more self-generated packets until it has earned enough credits by forwarding packets for others. Figure 2.8 shows some forwarding rules designed and investigated in [72] for implementing this approach. Here, *B* and *C* are initial battery power value and *nuglet* counter value of the node, and *N* represents the estimated number of intermediate nodes needed to reach the destination. *f* indicates the number of forwarded packets sent so far and *c* is the *nuglet* counter value at that point in time. \((NB-C)/(N+1)\) is the computed threshold based on which various candidate rules are applied [72]. The four rules studied are summarized in Figure 2.8. Of these, the most cooperative rule is Rule 1 while Rules 2, 3 and 4 are less cooperative, in that order.
Chapter 2 Literature Review

Figure 2.8. Forwarding Rules for the Nuglet Currency System

| Rule 1: | if \( f < \frac{(N-B-C)}{(N+1)} \) then forward  
|         | else drop |
| Rule 2: | if \( f < \frac{(N-B-C)}{(N+1)} \) then  
|         | if \( c \leq C \) then forward  
|         | else forward with probability \( \frac{C}{c} \) or drop with probability \( 1 - \frac{C}{c} \) |
|         | else drop |
| Rule 3: | if \( f < \frac{(N-B-C)}{(N+1)} \) then  
|         | if \( c \leq C \) then forward  
|         | else drop |
|         | else drop |
| Rule 4: | if \( f < \frac{(N-B-C)}{(N+1)} \) then  
|         | if \( c \leq C \) then forward with probability \( 1 - \frac{C}{c} \) or drop with probability \( \frac{c}{C} \)  
|         | else drop |
|         | else drop |

One disadvantage of this scheme is that additional overheads are required in the form of cryptographically protected security headers so that the users cannot tamper with their respective nuglet counters. (Such tampering will defeat the basic purpose of the scheme.) A trusted and tamper resistant hardware module must be present in each node as well. Another disadvantage is that a node which happens to be placed in an unfavorable position may not be able to earn enough nuglets to allow it to transmit its own packets. For example, this may happen if the node is placed near the boundary of the service area and most other nodes do not consider it a suitable choice as an intermediate node for routing their packets. The Nuglet system here focuses only on developing a packet forwarding strategy. Routing strategies like route discovery and route maintenance processes of on-demand routing protocols have not been addressed.

*SPRITE* (a Simple, Cheat-Proof, Credit-based System for Mobile Ad-hoc Networks) [74] does away with the need of a tamper resistant hardware module to be present at each node. A trusted central charging and payment system called Credit Clearance Service (CCS) is employed to handle virtual money or credit deduction and payment matters instead. When a node receives a message, it keeps a receipt of the message.
Later, when the node has a fast connection to a CCS, it reports to the CCS the messages that it has received/forwarded by uploading its receipts so as to receive payment. The payment and charges are determined from a game theoretical perspective, and it was shown that the game played is cheat-proof, meaning that telling the truth is the optimal strategy for each node. The game is also collusion resistant in the sense that a group of colluding players can find no better gains from playing other strategies, except by telling the truth. The disadvantage here is that a central server (CCS) is required. This may be a somewhat unreasonable demand in an ad hoc network as a CCS may not be feasible in ad hoc situations. Even in a situation where a CCS does exist, its continued availability and connection to all nodes will be crucial to the operation of this scheme and that will also be a difficult thing to ensure in a mobile and ad-hoc network.

For a system using SPRITE, Figure 2.9 (taken from [74]) shows the message success rates for two ad hoc networks - one network with 70 nodes uniformly distributed in a 1000m x 1000m area and another with 200 nodes uniformly distributed in a 2000m x 2000m area. Message success rate is defined as the percentage of messages that are successfully delivered from the sender to the destination. The communication radius of each node is 250m. In this experiment, since the nodes are power-and-credit-conservative, their estimated credit balance $c$ is close to 0 and their initial credits are chosen from a uniformly distributed range of $[0, C]$, where $C = 10$. To observe the effect of the amount of node resource on the overall message success rate, for each node, the number of messages that can be sent/forwarded by the remaining battery power of the node, as indicated by $b$, is chosen from a uniformly distributed range $[0, B]$, where $B$ is from 30 to 640. In other words, $b$ indicates the remaining battery power of a node, which is just enough to send out $b$ messages. An approximate analytical
expression for the message success rate is given in [74] as \(1 - \left(1 - \frac{C+1}{2BL}\right)^L\), where \(L\) is the average number of path hops. Using this as a reference plot simulation results are also generated for comparison. Figure 2.9 shows that with increasing values of resource \(B\), the nodes are more willing to forward others’ messages. In such a situation the message success rate is very close to 1.

Figure 2.9. Message Success Rate vs. Network Battery Resource
(extracted from [74])

CORSAC [75]-[76], a Cooperation-optimal Routing-and-forwarding protocol in wireless ad-hoc networks using cryptographic techniques, employs cryptographic techniques to deter nodes from cheating in a reward-based system. The protocol prevents cheating in the routing and forwarding stages. In the routing stage, a cryptographic technique is designed and implemented so that all link costs are reported truthfully to a destination. This allows the destination to select the correct minimum cost path eventually. Link cost is a function of the transmission power required between the two nodes involved. In the forwarding stage, data packets are transmitted in blocks. The destination needs to confirm the receipt of a block by sending the intermediate nodes a confirmation, before the intermediate nodes can proceed to transmit the next block of data packets. Block confirmation is implemented using reversed hash chain.
CORSAC focuses on designing and implementing strong system security measures to enforce routing decisions and following such a routing decision is the optimal action of each node in the sense that this brings maximum utility to the node. Forwarding and routing in a reward-based system are also addressed in this thesis. However, strong system security measures are assumed to be in place and the focus here is on developing forwarding and routing protocols to improve system performance. In Reward-based systems, the actions of others do not directly affect the action of an individual. However, if some nodes decide not to forward for others then they will deplete their own credits and will eventually be unable to transmit their own self-generated packets as well. This self-regulatory nature where a node can regulate its forwarding and self-transmission traffic by monitoring its own credit level makes it a good cooperation-enforcement approach. This is the reason why this approach is followed in the schemes proposed later in Chapter 4.

2.4 Multi-objective Routing

Various researchers have designed routing protocols that consider more than one routing factors or metrics. Some of them make use of available location information (obtained from GPS) to help them in making the routing decision. Reference [29] makes use of an asymmetric, exponentially-weighted forwarding function of forwarding distance, consumed energy and residual energy for forwarding. Given that $N'_c$ is the candidate set that includes the all neighbors of node $c$, each of which has a shorter distance towards the destination node $d$ than node $c$. The forwarding function in the case of node $c$, with respect to a node $n_i \in N'_c$, is given by:
Chapter 2 Literature Review

\[ f_c(n_i) = e^{\frac{\text{Gain}(n_i)}{\text{MaxGain}}^{c}} + e^{\left(1 - \frac{E_c(n_i)}{\text{MaxE}_c}\right)^{c}} + e^{\left(1 - \frac{E_l(n_i)}{\text{MaxE}_l}\right)^{c}} \]  

(2.13)

where \( \text{Gain}(n_i) = \text{Dist}(c, D) - \text{Dist}(n_i, D) \) is the difference between the distance from node \( c \) to node \( d \), and the distance from node \( n_i \) to node \( d \). This is known as the forwarding achievement.

\( E_c(n_i) \) is the consumed energy of node \( n_i \).

\( E_l(n_i) \) is the residual energy of node \( n_i \).

\( \text{MaxGain}, \text{MaxE}_c \) and \( \text{MaxE}_l \) are the maximum forwarding achievement, maximum consumed energy and maximum residual energy respectively observed among all nodes in \( N'_c \).

The aim of a node \( c \) is therefore to find the next hop node to deliver a packet towards the destination node with the objective as \( \max_{n_i \in N'_c} f_c(n_i) \). Periodic beaconing is required to disseminate the energy and position information for nodes to allow them to compute and make the forwarding decisions. Greedy forwarding is used to deliver data packets and route discovery is not required. However, recovery from greedy forwarding failure is required.

Reference [30] proposes various localized routing algorithms where nodes make routing decisions solely on the basis of the locations of their neighbors and the location of the destination. Its power-aware localized routing algorithm attempts to minimize the total power needed to route a message between a source and a destination. Its cost-aware localized routing algorithm aims to extend the worst-case battery lifetime at each node. Its combined power-cost localized routing algorithm attempts to minimize the total power needed while avoiding intermediate nodes with short residual battery remaining.
lifetime. Figure 2.10 shows the power-cost localized routing algorithm executed at each node $B$ when it needs to forward a packet to a possible node $A$ in the direction towards destination node $D$. Node $S$ is the source node.

![Power-cost-routing(S,D):](image)

One possible power-cost function is $\text{power-cost}(B,A) = f(A)u(r)$, $f(A) = 1/ g(A)$ where $g(A)$ is the remaining battery lifetime of node $A$ and $u(r)$ is the transmission power required for forwarding a packet from node $B$ to node $A$. $v(s)$ is the estimated transmission power required for forwarding a packet from node $A$ to node $D$. $f'(A)$ is the average value of $f(A)$ and all $f(X)$ values, where $X \in N_A$ and $N_A$ is the set of neighbors of node $A$. All the algorithms are executed at individual nodes and on a per data packet basis. No route discovery is required but loop avoidance needs to be enforced for the data packets.

Reference [77] designs various on-demand routing protocols that address max-min remaining energy and min-max link power routing problems. These protocols do not need location information but require route discovery for route determination. The Hybrid adaptive Energy-Aware Routing (HEAP) protocol performs routing by considering the battery power and transmission power required of the nodes in each route discovered. A route is chosen if (a) a source node and each forwarding node can
reach their next downstream neighbors using a selected transmission power value, and (b) the battery power level of each forwarding node is above the battery power threshold determined by the source. In version 1 (HEAP-1), if the source node does not receive any reply, it will initiate another route discovery by decreasing the battery power threshold. It will repeat this process until the battery power threshold is at its lowest where all nodes should be included in the search. If the source node fails to get any replies at this point, it will increase the transmission power threshold in the next route discovery initiated. The search will continue with increasing transmission power threshold set, until this threshold value approaches the maximum possible transmission power level or a route has been found. The initial transmission power threshold level of a node can be obtained from the transmission power threshold value carried in the RREQs passing through it. In version 2 (HEAP-2), if the source node does not receive any reply, it will initiate another route discovery by increasing the transmission power required. It will repeat this process until all nodes are transmitting with the maximum transmission power. If the source node fails to get any replies at this point, it will decrease the battery power threshold in the next route discovery initiated. To determine the global battery power threshold, each node will broadcast its battery power level if its level drops to the next lower threshold. If the battery level of a certain percentage of nodes has dropped to the next lower threshold, the global battery power threshold will be adjusted.

In [31], the proposed Positional Attribute based Next-hop Determination Approach (PANDA) uses positional attributes to determine the rebroadcast delay of each forwarding node during the route discovery phase. Such attributes include relative distance between two intermediate nodes, estimated link lifetime, transmission power
consumption and residual battery power. A forwarding algorithm is developed where each attribute is considered individually. This is then integrated into an on-demand routing protocol for routing. In [77], a performance comparison of PANDA-TP (transmission power) is made with the earlier work in [32]. PANDA-TP is based on the fact that for a given distance, increasing the number of hop counts will decrease the total transmission power consumption. Therefore, the rebroadcast delay will be shorter if the distance between the upstream neighbor (immediate sender of route request packet) and the forwarding node concerned is shorter. Figure 2.11 shows the algorithm that determines the rebroadcast delay for a node A when it receives a RREQ from a upstream node S in the case of PANDA-TP. \(L_1, L_2\) and \(L_3\) are distance threshold values that satisfy the relationship \(L_3 < L_2 < L_1\). \(t_1\) is the base delay time and \(\text{uniform}(0, t_1)\) will return a random value uniformly distributed between 0 and \(t_1\). The destination will wait for a period of time to collect enough route information, before deciding on the route that the source should use.

\[
\text{At node } A: \\
\text{if } \|SA\| < L_3 \\
\text{delay} = t_1 + \text{uniform}(0, t_1) \text{ //this is Class 1} \\
\text{else if } \|SA\| < L_2 \\
\text{delay} = 2* t_1 + \text{uniform}(0, t_1) \text{ //this is Class 2} \\
\text{else if } \|SA\| < L_1 \\
\text{delay} = 4* t_1 + \text{uniform}(0, t_1) \text{ //this is Class 3} \\
\text{else} \\
\text{delay} = 6* t_1 + \text{uniform}(0, t_1) \text{ //this is Class 4}
\]

Figure 2.11. Determining Rebroadcast Delay in PANDA-TP

Figure 2.12 shows the graph of Network Lifetime against Mobility Ratio extracted from [77]. Only the results of the better performing protocols from [77], HEAP-1 and HEAP-2, are shown here, comparing with PANDA-TP (from [32]) as chosen by the authors of
Network lifetime is defined as the time till the first node runs out of battery power. Mobility ratio is the ratio of total moving time to total simulation time. The velocity of node movement is randomly selected between 1 and 20 metres per second. The mobility ratio is changed by varying the stationary time. Figure 2.12 shows that HEAP-2 and HEAP-1 outperform PANDA-TP. While PANDA-TP consistently chooses routes that consume as little transmission power as possible, it does not consider the current battery power status of the forwarding nodes involved. HEAP-1 enhances its pure battery power version by excluding those unnecessary extra-long links. HEAP-2 enhances its pure transmission power aware version by filter out those energy-critical nodes from participating in multi-hop data transmissions. In all cases, network lifetime performance of each of the protocols increases with mobility ratio. This is because mobility helps to balance the traffic distribution among nodes and avoid certain nodes to be over drained by having to serve as relaying nodes for many connections too long. In this thesis, the updated version of PANDA-TP (transmission power) from [31] and the better performing HEAP version 2, HEAP-2, as claimed in [77] will be considered for performance comparisons in Chapter 5.

![Figure 2.12. Graph of Network Lifetime against Mobility Ratio (extracted from [77])](image-url)
In [78], a simple routing scheme where each node, that wants to transmit toward the destination, makes a local routing choice selecting its neighbour node to minimize the targeted multi-objective function. The multi-objective function used considers transmission power, node battery power and link stability. Each node has to process and make individual decision on a per data packet basis. Similar multi-objective functions are formulated in Chapter 5. However, there is additional objective here, i.e. to design practical on-demand multi-objective routing protocols for large and mobile networks. The performance comparisons will be made with \textit{HEAP-2} and \textit{PANDA-TP} as these are suitable for routing in large and mobile networks.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

3.1 Introduction

A technique for power conservation and power-aware routing is proposed and integrated in the Ad-hoc On-demand Distance Vector (AODV) protocol. As mentioned in Section 2.1, AODV is a popular and well established choice for current wireless ad hoc network implementations and simulation studies and the effort here is to propose the addition of power-aware features to AODV in a manner which can be easily integrated with the current AODV standards. The proposed scheme and its performance results are presented in detail in this chapter. These results are compared with the performance of a system using the original (unmodified) AODV protocol. It should be noted that the focus is on AODV rather than DSR for these proposed changes since (as indicated in Section 2.1) AODV is more scalable and better suited for large scale mobile network implementations than DSR.

The proposed Power Aware AODV (PAW-AODV) protocol is described in Section 3.2. A more detailed description of the PAW-AODV Route Discovery, Route Maintenance and Cost Maintenance processes are presented in Sections 3.3, 3.4 and 3.5, respectively. The main differences between AODV and PAW-AODV are summarized and highlighted in Section 3.6. The performance results obtained from stationary and mobile scenarios are presented and discussed in Section 3.7. Section 3.8 presents the conclusions of this chapter.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

3.2 Protocol Description

The proposed PAW-AODV retains the basic features of the AODV algorithm [36]. PAW-AODV is still an on-demand routing algorithm and only records a single route entry for each destination in the route tables. In fact, the sequence number and timer-based states techniques suggested for AODV are also employed. The power aware techniques incorporated, which constitute the main features of PAW-AODV may be summarized as follows. Instead of considering hop count, a cost function based on the available battery power of the individual nodes along a particular path is used as the route cost parameter for choosing that path. The cost function of the route is the sum of the cost functions of the individual nodes along the route where the cost function of a node is dependent on the node’s available battery power. Using this definition of the route costs, the aim of PAW-AODV is to incorporate appropriate route finding algorithms in AODV to find a route $r$ with the least cost. For example, a simple node and route cost function of this type may be defined as in (3.1) and (3.2) respectively. Here $c_i(t)$ is the cost function of node $i$ at time $t$ where $b_i(t)$ is the remaining battery power of node $i$ at time $t$. $r$ is the set of intermediate nodes located along this particular route. Thus, the route cost $c(r,t)$ of route $r$ at time $t$ is the summation of the node costs of all nodes located along this route as given by (3.2).

\[
c_i(t) = \frac{1}{b_i(t)} \quad \text{(3.1)}
\]

\[
c(r,t) = \sum_{i \in r} c_i(t) \quad \text{(3.2)}
\]

The node cost function and the route cost function given in (3.1) and (3.2) are merely indicative of the approach that may be taken if power aware route costs are calculated.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

The actual forms of the cost functions used in the proposed PAW-AODV are elaborated in detail in Section 3.5.

3.3 Route Discovery

As in the case of AODV, the route discovery process is initiated whenever a source node needs to communicate with another node for which it has no routing information in its table. The source node then broadcasts a route request (RREQ) control packet to its neighbors. The RREQ is augmented to include a $T_{crRREQ}$ field, in addition to the original fields contained in RREQ. This $T_{crRREQ}$ field in the RREQ packet is meant to be a depository of the route cost as the RREQ packet propagates in the network. The initial value of this field is zero (when the RREQ is launched by the source for route discovery) and its content is built up node by node as the sum of the costs of the nodes through which the RREQ passes as it propagates in the network.

In PAW-AODV, when an intermediate node receives a RREQ it has seen at least $L$ times before (i.e. a RREQ with the same broadcast_id and source_addr) within the same flooding interval, it drops the RREQ and does not process it. Recall that the pair $<source_addr, broadcast_id>$ uniquely identifies a RREQ. This is because broadcast_id is incremented whenever the source issues a new RREQ, and the source_addr uniquely identifies the sender’s address. The flooding interval is the time interval in which a particular source broadcasts a RREQ and then waits for the intended destination to reply. The flooding interval is defined as the time taken for a RREQ to reach the destination and for the corresponding RREP to return to the source [36]. (Note that $L$ is a parameter that needs to be chosen carefully. The equivalent choice in classical AODV
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

is $L=1$.) Allowing a choice of $L>1$ allows more routes of possibly lower route cost values to be discovered in PAW-AODV. At the same time, keeping $L$ small is also desirable as it will limit the number of RREQ that will be processed and transmitted. It should be noted that even though RREQ transmissions are short, they do consume precious power resources of nodes and bandwidth resources of links and may cause packet collisions with other transmissions. This is the reason why excessive RREQ transmissions are undesirable.

Figure 3.1 illustrates the route discovery process of the PAW-AODV protocol. The arrows show the paths traveled by the RREQs originating from source node A. All the other nodes are intermediate nodes. The number beside each node indicates individual node cost – for this example, this is merely an indicative figure which may be calculated using a procedure like that of equation (3.1). Node F may receive a RREQ from node D first (probably because of the short distance of route $A \rightarrow D \rightarrow F$). However node F cannot reject the RREQ from node C within the flooding interval, since the route cost of $A \rightarrow C \rightarrow F$ is 3, which is smaller than the route cost of 4 of $A \rightarrow D \rightarrow F$. Node F may also like to see the RREQ from node E even though this RREQ traverses through more hops and may come in later; this is because, in this case, the route cost of 2 of $A \rightarrow B \rightarrow E \rightarrow F$ is even lower. However, it is necessary to set a limit on the number of RREQ per source that can be accepted within a flooding interval, so as not to cause over-flooding. The choice of this limit $L$ will be discussed further in this chapter.

Upon receiving a RREQ, a node either sends a route reply (RREP) back to the source or rebroadcasts the RREQ to its own neighbors. It sends a RREP back to the source if it has a path to the destination which is not outdated. Otherwise (i.e. if it has no path or if
the path is outdated), it re-broadcasts the RREQ to its own neighbors but modifies the RREQ by incrementing the hop count or $hop\_cnt$ field and adding its node cost value to the value contained in the $Tcr_{RREQ}$ field of the received RREQ. The Reverse Path Setup and Forward Path Setup processes in PAW-AODV are described next.

![Figure 3.1. RREQ Handling in PAW-AODV](image)

### 3.3.1 Reverse Path Setup

There are two sequence numbers included in a RREQ. These are the source sequence number and the last destination sequence number known to the source. The source sequence number is used to maintain updated information about the reverse route back to the source, and the destination sequence number reflects how updated a route to the destination must be before it can be accepted by the source. As the RREQ travels from a source to various intermediate nodes before finally reaching its destination, it automatically sets up the reverse path from all these nodes back to the source. If a reverse path already exists, it is updated when either of the following conditions is met:

(a) the source sequence number $source\_sequence\_#$ in the RREQ is higher than the destination sequence number $dest\_sequence\_#$ of the corresponding reverse route entry in the node’s route table,
or (b) the sequence numbers are equal, but the route cost $T_{cr\text{RREQ}}$ as specified by the RREQ is lower than the route cost $T_{cr\text{RoutingTable}}$ currently present as the cost of the corresponding reverse route entry in the node’s route table.

It may be noted that in the proposed PAW-AODV protocol, $T_{cr\text{RREQ}}$ is recorded along with the other AODV route information during the reverse route setup or update process. In this instance, the $T_{cr\text{RoutingTable}}$ field in the route table is updated with the $T_{cr\text{RREQ}}$ value.

### 3.3.2 Forward Path Setup

If an intermediate node’s route table has no route to the destination node or if the route in the route table is outdated, then the intermediate node will broadcast RREQ with its own node’s routing cost (i.e. its $c_i(t)$ value) included in the $T_{cr\text{RREQ}}$ of the new RREQ. On the other hand, if the intermediate node does have a current route to the destination node which is suitably updated, it sends back a RREP to the source node. The total route cost from the source to destination $T_{cr\text{RREP}}$ is quoted, together with the other AODV fields, in the RREP. The total route cost from the source to destination $T_{cr\text{RREP}}$ includes the $T_{cr\text{RREQ}}$ value (reverse route-to-source cost), the $T_{cr\text{RoutingTable}}$ value (forward route-to-destination cost) of the forward route and the node cost $c_i(t)$. Similarly, a gratuitous RREP [36] is sent to the destination node if the corresponding RREQ has the ‘G’ (Gratuitous) flag set. The total route cost from source to destination $T_{cr\text{RREP}}$ is also quoted, together with other AODV fields, in the gratuitous RREP.
If the destination receives RREQ from the source, it will send out RREP in a similar manner as discussed above, except that the destination sequence number of the RREP to be sent out is updated from the RREQ. If a Gratuitous RREP is received, the reverse route to the source is created if the destination’s route table does not have an entry; otherwise, the earlier entry is updated if either of the following conditions is met:

(a) the destination sequence number (which is actually the source sequence number) in the Gratuitous RREP is greater than the corresponding destination sequence number of the reverse route entry in the node’s route table,

or (b) the sequence numbers are the same, but the earlier route is no longer active (due to active route timeout mechanism whereby a route not used for more than a timeout interval as defined in [36] will be expired) or the Gratuitous RREP carries a lower total route cost $T_{crGRREP}$.

Since the reverse route has already been established by the RREQ traveling from the source to the node issuing the RREP, the RREP can travel back to the source along this route. As it does so, each node along the reverse path sets up a forward pointer to the node from which the RREP came from, updates its timeout information for route entries to the source and destination, and records the latest destination sequence number for the requested destination.

Regardless of the type of RREP packets received, the forward route for the destination is created if the node’s route table does not have an entry; otherwise, the entry is updated only if either of the following conditions is met:
(a) the destination sequence number in the RREP is greater than the corresponding destination sequence number of the route entry in the node’s route table,

or

(b) the sequence numbers are the same, but either the earlier route is no longer active or the RREP carries a lower total route cost value from this intermediate node to the destination.

The forward route cost field \( T_{cr}^{RoutingTable} \) of the route table is updated appropriately. In addition to the total route cost field \( T_{cr}^{RREP} \), each RREP also carries an accumulated route cost field \( A_{cr}^{RREP} \), for the purpose of the computation of \( T_{cr}^{RoutingTable} \). Consider Figure 3.2 as an example, to explain this computation. The arrows show the path to be traveled by RREP from node D (destination) to node A (source). The numbers beside the nodes show the node cost \( c_i(t) \). As node D receives the RREQ with route cost information (which is \( T_{cr}^{RREQ}=4 \)), it will set both \( T_{cr}^{RREP} \) and \( A_{cr}^{RREP} \) to be equal to \( T_{cr}^{RREQ} = 4 \) and then sends out the RREP to node C. As node C receives the RREP, it will compute \( T_{cr}^{RoutingTable} \) and adjust \( A_{cr}^{RREP} \) based on (3.3) and (3.4) respectively. So node C will set \( T_{cr}^{RoutingTable} = 4 - 4 = 0 \), indicating a zero route cost to node D, as it can send data directly to node D and hence no forwarding cost is incurred. Then node C sets \( A_{cr}^{RREP} = 4 - 3 = 1 \). Likewise, upon receiving the RREP from node C, node B will set \( T_{cr}^{RoutingTable} = 4 - 1 = 3 \), indicating a route cost of 3 units is required for node B to route to node D via node C. Then node B sets \( A_{cr}^{RREP} = 1 - 1 = 0 \). Finally, when node A receives the RREP, it sets \( T_{cr}^{RoutingTable} = 4 - 0 = 4 \), indicating a total route cost of 4 units is required for node A to route to node D via node B and C. No further action is required on \( A_{cr}^{RREP} \) as the source has been reached. If node D is an intermediate node
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

that has forward route information, $Tcr_{RREP}$ will be set to the total route cost from source to destination (which will be a node downstream of node D) instead.

![Path traveled by RREP](image)

$Tcr_{RREP} = 4$
$Acr_{RREP} = 0$

$Tcr_{RREP} = 4$
$Acr_{RREP} = 1$

$Tcr_{RREP} = 4$
$Acr_{RREP} = 4$

$Tcr_{RREP} = 4$
$Acr_{RREP} = 4$

$Tcr_{RoutingTable} = Tcr_{RREP} - Acr_{RREP}$  \hspace{1cm} (3.3)

$Acr_{RREP} = Acr_{RREP} - c_i(t)$  \hspace{1cm} (3.4)

The RREP is relayed upstream until it finally reaches the source. The source node will begin data transmission as soon as the first RREP is received, but it can modify its route information (and change the routing) if it learns of a better route later.

3.4 Route Maintenance

If nodes which are not along any active path move, nothing needs to be done as no data transmission was occurring at their previous locations. If the source node moves during an active session and still needs to transmit to a destination node, it can reinitiate the route discovery procedure to establish a new route to this destination. This is because some or all of the existing intermediate nodes may not be able to route based on the new location of this source node. When either the destination or some intermediate node
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

moves, a typical AODV Route Error (RERR) control packet is sent to the affected source nodes.

Once the next hop becomes unreachable, the intermediate node directly affected by the link break propagates a RERR to the list of precursor nodes also affected by the link break. The affected destinations and their respective destination sequence numbers are recorded in the RERR before the RERR is propagated upstream. If there is only one precursor node for the affected destination, RERR is unicast to this precursor node; otherwise, the RERR is broadcast further. When the neighbors receive the RERR, they will mark their routes to the affected destinations as invalid by setting the distance to the destination equal to infinity and will then in turn propagate the RERR to their precursor nodes, if any. These entries are not deleted immediately because they may still contain useful routing information. Instead, they expire in a time frame approximately equal to the expiry time of the reverse route.

Once the source receives the RERR, it can reinitiate RREQ if the corresponding route is still required. To rediscover the route again, the source will send out an RREQ with a destination sequence number one greater than the previously known sequence number. This is to ensure that it builds a new, viable route, and that no nodes reply if they still regard the previous route as valid. RERR is also generated and broadcasted when a node receives a data packet destined for another node but it does not have an active route to that node. Thus, the upstream neighbors can be informed not to send any more data packets to the node which no longer has a route to the destination.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

3.5 Cost Maintenance

3.5.1 Basic Concept

In addition to the Route Maintenance process that handles link failure, PAW-AODV also implements a Cost Maintenance process that adjusts the node’s cost, which will affect the route cost, according to the node’s available battery power. The new node cost will be announced by broadcasting Route Cost Announcement (RRCA) packets to the list of precursor nodes affected by this cost change. This is analogous to and is motivated by what AODV does with its RERR packets when a link break is detected. In fact, RRCA is similar to RERR, except that it has an additional $\text{diff}_c^{RRCA}$ field to record the cost difference. Note that if there is only one precursor node, RRCA will be unicast only to that node.

Consider Figure 3.3 to illustrate how RRCA gets propagated and handled when node D’s cost changes from 3 to 4 (say). For this $\text{diff}_c^{RRCA}$ is set to 1 by node D, which is the difference between new node cost (4) and old node cost (3). It will broadcast RRCA to node B and node C (i.e. nodes in the precursor list of node D) that are routing via node D to node F, the final destination. Node E need not be informed since it is not routing via node D. Node B will update the route cost (from node B to node F) with this new information, adding $\text{diff}_c^{RRCA} = 1$ to original route cost (3) to get new route cost (4). It will then unicast this RRCA to node A since the latter is the only node routing via node B (and node D) to node F. In the same way, node A will add $\text{diff}_c^{RRCA} = 1$ to original route cost (4) to get the new route cost (5). Node C (and node A as well) need not send out the RRCA since there are no nodes upstream that need this information. Node A and
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

C can then decide whether to rediscover the route again if the route cost is now higher. If the new route cost is lower, then they can continue to use the route.

![Diagram of RRCA Propagation and Handling in PAW-AODV](image)

Figure 3.3. RRCA Propagation and Handling in PAW-AODV

### 3.5.2 Practical Design

If RRCA is sent every time a node’s cost changes with changes in its available battery power as reflected in (3.1), then the control traffic generated due to the route announcement process will be overwhelming. The node cost will hardly have a chance to stabilize because of the frequent changes caused by the constantly decreasing available power at the node. This is clearly impractical and led to the proposal of a cost zoning concept for this purpose. This concept defines a few zones based on a specific range of the node’s available power and then assigns a fixed node cost to each zone. The node’s cost does not change as long as the zone boundary is not crossed. This is expressed in (3.5).

\[
c_j(t) = \begin{cases} 
CR_1 & \text{for } b_j(t) > B_1 \\
CR_2 & \text{for } B_2 < b_j(t) \leq B_1 \\
CR_3 & \text{for } B_3 < b_j(t) \leq B_2 \\
\vdots & \text{for } B_i < b_j(t) \leq B_{i-1} \\
CR_j & \text{for } B_j < b_j(t) \leq B_{j-1} 
\end{cases} 
\]  

(3.5)
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

In (3.5), there are $j$ steps (or zones) shown. In each zone, a cost $CR$ can be assigned based on the range of available battery power, where $B_1, B_2, B_3, \ldots, B_j$ each represents a percentage of the initial full battery power with $B_1 > B_2 > B_3 > \ldots B_{j-1} > B_j$. Retaining an inverse relationship as in (3.1), $CR_1 < CR_2 < CR_3 < \ldots CR_{j-1} < CR_j$ in (3.5). While testing the proposed algorithm through simulations, it was discovered that RRCA does contribute significantly to the total traffic and there is a need to control the number of RRCA packets being sent. On an experimental basis, four cost zones are configured based on a node’s initial battery power as shown in Table 3.1.

Table 3.1. Cost Zone Table

<table>
<thead>
<tr>
<th>Zone</th>
<th>Node’s available power as a percentage of initial power</th>
<th>Assigned node cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>&gt; 30% of initial battery power</td>
<td>$CR$</td>
</tr>
<tr>
<td>Yellow 1</td>
<td>&gt; 20% but $\leq$ 30% of initial battery power</td>
<td>2*$CR$</td>
</tr>
<tr>
<td>Yellow 2</td>
<td>&gt; 10% but $\leq$ 20% of initial battery power</td>
<td>4*$CR$</td>
</tr>
<tr>
<td>Red</td>
<td>$\leq$ 10% of initial battery power</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

The important implications of the choice of four zones as in Table 3.1 may be noted as follow. When a node has sufficiently high power, PAW-AODV will tend to behave like AODV. This means, at this high power state or in the white zone as indicated in Table 3.1, the cost function is effectively equivalent to one hop count between immediately reachable neighboring nodes; this is the same behavior as in traditional AODV. When a node’s available power drops to the ‘Yellow’ alert zones, the cost function is forced to be set to a value which is more than one hop count value between the immediate reachable neighboring nodes. In this way, source nodes are discouraged from routing via the nodes that have entered the ‘Yellow’ alert zone because of the higher node costs that these nodes would have announced. One can have a number of ‘Yellow’ alert zones, depending on the number of power level thresholds that will be implemented.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

Here, only two ‘Yellow’ alert zones are implemented in the present simulation studies. The ‘Red’ alert zone is reached when the node’s available power is very low. A node in this zone will indicate a very high node cost to others to deter them from routing packets through it. Its intention here is to reserve the remaining power for its own usage (i.e. for transmitting the data packets that it generates).

Table 3.1 may be expressed mathematically as (3.6). Like (3.5), this expresses a step-wise relationship, instead of a continuous relationship, between the node cost and the node’s remaining battery power. Here $B$ is the initial full battery power value and $CR$ denotes a standard node cost unit that is used; $CR=1$ is conveniently assigned in this case. Figure 3.4 shows the relationship. In the Red Zone, the node cost is assigned a very high value essentially equivalent to infinity.

$$c_i(t) = \begin{cases} 
CR & \text{for } b_i(t) > 0.3B \\
2CR & \text{for } 0.2B < b_i(t) \leq 0.3B \\
4CR & \text{for } 0.1B < b_i(t) \leq 0.2B \\
\infty & \text{for } b_i(t) \leq 0.1B
\end{cases}$$  (3.6)

Figure 3.4. Step-wise relationship between node cost and remaining battery power
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

The simulation experiments indicated that, for the same reasons as the ones mentioned earlier, it would be desirable to reduce the control traffic to the extent possible. To aid in this, instead of the obvious approach of broadcasting RRCA packets, the following approach is proposed. On entering the Red zone, the node not only sets its node cost very high, but it also sends a special Power Aware PAW-RERR or low power RERR message to its affected precursors which use it for forwarding packets to their destinations. This is to alert all precursor nodes immediately so that they do not route any more packets via the affected low power node (as it will not forward any more packets for other nodes). Though the low power RERR is similar to the AODV RERR, there is one major difference in the action taken when this low power RERR is sent. PAW-AODV does not expire any routes in the low power RERR sender’s route table after the low power RERR is sent. [In contrast, AODV does expire routes affected by broken links that trigger its RERR transmission]. This is because it may still use this route to transmit self generated packets. Once the source receives the RERR, it can reinitiate RREQ if such a route is still required.

For the Yellow zones, the proposed PAW-AODV does not make a specific announcement (as suggested above for a node entering the Red zone) for this by transmitting a RERR kind of cost announcement. This can also be incorporated, if so desired. However, the simulation studies show that (a) these announcements were not particularly effective in improving the system’s performance and, moreover, (b) the extra traffic generated by these extra RERRs seems to have a significant detrimental effect on the throughput and delay performances.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

It may be noted that the number of zones, the associated node’s available power range and assigned node costs shown in Figure 3.4 are meant to be indicative of the proposed approach. The current approach used, as indicated in Figure 3.4, intends to let PAW-AODV behaves in the same manner as AODV for most of a node’s lifetime, as long as its current battery power level is greater than 30% of the initial power allocated (as in the White zone). When the current battery power level drops to be within 10% to 30% of the initial power allocated (as in the two Yellow zones), which is considered to be much lower than half of the initial power allocated, a node will discourage but not disallow other nodes to route via it. If other nodes are worse off, this node will have to forward the data packets. However, if current battery power level drops to below 10% of the initial power allocated (as in the Red zone), this power level is considered to be very low and the node should reserve the remaining power for its own use. Different and specific values may be chosen in an actual system so that the system may be suitably configured depending on actual values of its battery power and the performance that it is expected to provide. The values of Figure 3.4 are used for the simulation runs reported here, but these will be varied subsequently, as a future performance tuning work to be done, to see their effect on the system’s performance.

3.6 Protocol Comparison

Figures 3.5 and 3.6 show the modifications done to create PAW-AODV. Figures 3.5 and 3.6 show the Route Discovery and Route (Cost) Maintenance processes respectively, of both AODV and PAW-AODV. In Figure 3.5, the modifications made to the AODV Route Discovery process to create PAW-AODV are indicated by the bold remarks, placed next to the stage (of the process flow) that the modifications were carried out.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

Figure 3.5. Process flow of AODV and PAW-AODV Route Discovery
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

Figure 3.6. Process flow of AODV and PAW-AODV Route (Cost) Maintenance

(a) AODV Route Maintenance

- Battery power level drops
  - Different zone?
    - No action needed
    - Different zone?
      - Yellow zone?
        - Red zone entered – low power RERR is sent to affected list of precursors
  - Intermediate node?
    - Source initiates RREQ with a higher destination number if route is still required
    - Node sends RRCA to list of precursors affected by node cost change
    - Next node receives RRCA
      - Intermediate node?
        - Route cost now higher?
          - Continue using route
          - Route cost now higher?
            - Source initiates RREQ with a higher destination number if route is still required
            - Source sets route cost destination according to the diff_csrca field
            - Intermediate node?
              - Set route cost to destination according to the diff_csrca field and send RRCA to affected list of precursors
              - Set route distance to destination as infinity and send RERR to affected list of precursors

(b) PAW-AODV Cost Maintenance

- Node has detected link break
- Node sends RERR to list of precursors affected by break
- Next node receives RERR
- Intermediate node?
  - Node cost change only
    - Red zone entered – low power RERR is sent to affected list of precursors
    - Handling of low power RERR by precursor nodes is similar to the RERR handling described in Figure 3.6(a)
  - Source sets route cost destination according to the diff_csrca field
  - Route cost now higher?
    - Continue using route
    - Route cost now higher?
      - Source initiates RREQ with a higher destination number if route is still required
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

The handling of RERR is not included in Figure 3.5 as it is reflected in Figure 3.6 instead, where its involvement is in the Route Maintenance process. Figure 3.6(a) shows the AODV Route Maintenance process while Figure 3.6(b) shows the two phases of the PAW-AODV Cost Maintenance process. As mentioned in Section 3.5.2, RRCA was initially used (RRCA cost maintenance process flow (phase 1) is indicated by the non-bold boxes) after the decision of whether a different zone is entered has been made. Regardless of whether it is a Yellow or a Red zone, the same process flow is followed. When a low power RERR is used instead, the cost maintenance process flow (phase 2) is indicated by the bold boxes after the decision of whether a different zone is entered has been made. When Yellow zone is entered in phase 2, the nodes do change their node costs but no control packets (low power RERR) are sent out.

Overall, Table 3.2 summarizes the differences between AODV and PAW-AODV as well as the modifications made in AODV to implement the desired power-aware features in PAW-AODV. In the next section, both routing protocols will be put through various test scenarios to study their performance.
Table 3.2. Differences between AODV and PAW-AODV

<table>
<thead>
<tr>
<th>Features</th>
<th>AODV</th>
<th>PAW-AODV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost function</td>
<td>Based on hop count</td>
<td>Based on available battery power</td>
</tr>
<tr>
<td>Number of requests processed for each source</td>
<td>One within a flooding interval</td>
<td>More than one, if found, but subject to a limit $L$ within a flooding interval</td>
</tr>
<tr>
<td>Cost maintenance</td>
<td>Route Error notification mechanism is used to reflect route cost change when a node has moved away or a link is broken</td>
<td>In addition to using the AODV Route Error notification mechanism to handle node’s movement and broken link, PAW-AODV has a Cost Maintenance mechanism to change the node cost depending on the node’s available power level and inform affected precursor nodes when Red zone is entered</td>
</tr>
<tr>
<td>Data differentiation</td>
<td>No differential treatment between self generated and forwarded data packets</td>
<td>When node’s available power drops to the Red zone, only self generated packets will be transmitted</td>
</tr>
</tbody>
</table>

3.7 Performance Comparison

In this section, the performance of the proposed PAW-AODV is studied through simulations and its performance is compared to that of AODV under various power constrained scenarios. The network simulator used is QUALNET version 3.7 developed by Scalable Network Technologies Inc [79]. Qualnet is the commercial derivative of GloMoSim and has a wired & wireless network modeling library which makes simulations convenient to set up and run. Compared to GloMoSim, the additional features provided by QUALNET are (a) graphical user interface (GUI), (b) larger model library for wired and wireless networks and (c) built-in statistics collection and analysis capabilities. The details of the simulation set up and the results obtained are discussed in this section.
3.7.1 Simulation Environment

The simulation parameters for the test environment are summarized in Table 3.3. (The default QUALNET parameters left unchanged are indicated.) Two basic simulation scenarios, stationary and mobile, are considered here.

Stationary Scenario: This has 100 nodes placed in a 10x10 grid pattern as in Figure 3.7. The node separation (vertical or horizontal) is half the radio range, i.e. 97.5m. Therefore, a node may at most reach 12 other nodes directly (i.e. if it is in the centre), and a minimum of 5 other nodes directly (i.e. if it is at a corner), as shown in Figure 3.7. This kind of scenario may be expected in offices, static gaming environments, or in military scenario with wireless sensors placed at regular intervals. In the case here, this scenario aids in the understanding of the PAW-AODV and AODV so that the protocol specific parameters can be better tuned later.

Table 3.3. Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test area</td>
<td>2000m X 2000m</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>100</td>
</tr>
<tr>
<td>Placement</td>
<td>Uniform 10 X 10 (grid)</td>
</tr>
<tr>
<td>Node separation</td>
<td>Half of radio range, vertically and horizontally</td>
</tr>
<tr>
<td>Radio range</td>
<td>195m [Qualnet default]</td>
</tr>
<tr>
<td>Transmission Bandwidth</td>
<td>2Mbps [Qualnet default]</td>
</tr>
<tr>
<td>Antenna type</td>
<td>Omni directional antenna [Qualnet default]</td>
</tr>
<tr>
<td>MAC</td>
<td>IEEE 802.11 [Qualnet default]</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random waypoint [Qualnet default]</td>
</tr>
<tr>
<td>Mobility speed</td>
<td>0 – 10 m/s [Qualnet default]</td>
</tr>
<tr>
<td>Mobility pause time</td>
<td>30s [Qualnet default]</td>
</tr>
<tr>
<td>Traffic type</td>
<td>Bursty Source (see below), UDP</td>
</tr>
<tr>
<td>Number of sources</td>
<td>10 per minute</td>
</tr>
<tr>
<td>Packet size</td>
<td>64 bytes</td>
</tr>
<tr>
<td>Inter-arrival time</td>
<td>5ms</td>
</tr>
<tr>
<td>Packet transmission probability</td>
<td>0.1</td>
</tr>
<tr>
<td>Duration per voice connection</td>
<td>15s</td>
</tr>
</tbody>
</table>
Mobile Scenario: In this, 100 nodes are randomly placed in an area 2000mX2000m square. The nodes move in random directions using the random waypoint model with a speed randomly chosen (uniformly) in the range of 0-10 m/s with a pause time of 30s.

Traffic Generator: This is taken to be similar to a bursty sample traffic from [80]-[81]. In this, during every minute of the simulation, 10 source nodes are randomly chosen to set up a UDP connection with another 10 randomly chosen destinations. The transmissions for each of these connections start at random instants within this one-
minute interval. The connection for each source-destination pair lasts for 15s ($T$). Within this 15s interval, a packet of 64 bytes is generated once every 5ms ($1/\lambda$) with a probability of 0.1 ($P_{PacketTransmit}$). In effect, a source will send 300 packets (i.e. $\frac{T}{1/\lambda} \cdot P_{PacketTransmit}$) in each such connection to its destination.

To study the performance under power-constrained scenarios, limited amount of battery power was assigned to each node at the start of the simulations. It may be noted that a node exhausts battery power in proportion to the bits transmitted, both for transmitting/forwarding data packets and also for the transmission/forwarding of the overhead packets (RREQ, RERR etc.). However, no power is assumed to be spent otherwise (i.e. during internal processing). The overhead packets (RREQ, RERR etc.) will also consume power from this initial value; however, since these are smaller in size, their power consumption will be correspondingly lower than that of the data packet. (This is approximately in the ratio of 1:3 for the current choice of parameters). It should also be noted that the resource required by these overhead packets is not limited to the power they consume in transmission. Since the transmission of these packets is also done following IEEE 802.11 CSMA/CA MAC protocol, these may cause other nodes in range to defer their data packet transmissions (and vice versa).

Two choices for the initial power, uniform and random, were experimented in various simulation situations. For the *uniform initial power distribution* case, each node is initialized with enough power to transmit/forward a total of 4500 packets of traffic. In the *random initial power distribution* case, the nodes are randomly assigned a value of initial power uniformly distributed in the range such that it can transmit/forward a total of (450 – 4500) packets. However, only the results based on *uniform initial power*
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

distribution case are shown in this report as the results generated using random initial power distribution exhibit similar trends as the results generated using uniform initial power distribution. Moreover, the uniform initial power distribution case is likely to be a more common scenario. However, the results discussed and the insights gained apply equally well to both cases.

3.7.2 Stationary Scenario Results and Discussion

In Figure 3.8, results are presented to compare the basic AODV protocol with the proposed PAW-AODV approach for the stationary simulation scenario. It may be noted that the PAW-AODV scheme, whose performance is shown in Figure 3.8, corresponds to one which incorporates the Low Power Cost Maintenance Module using $L=3$ with thresholds as shown in Figure 3.4 and algorithm (Phase 2) as in Figure 3.6(b). The first graph Figure 3.8(a) shows the cumulative number of data packets received as a function of the time of operation of the system, i.e., the number of packets received at time $t$ is the total number of data packets that reach their respective destinations. The second graph Figure 3.8(b) shows the remaining battery power in the nodes of the system as a function of time. This is to indicate the extent to which the two protocols are able to consume the initial power that is provided when the system starts operation. The third and last graphs Figure 3.8(c) and (d) show the delay statistics as measured around the crossover point (5400s) and at the end of the simulation run (14000s) respectively. The minimum, maximum and mean delays encountered by packets until the indicated time, and the SQV of the delay ($= \text{variance}/(\text{mean})^2$) are shown in each of the delay statistics graph.
As shown in Figure 3.8(a), the proposed PAW-AODV protocol provides significant operational advantages compared to the original AODV protocol as it is able to convey more packets to their destination than AODV. At the initial stages (i.e., $t \leq 500$ s in this example), most of the nodes have enough power. In this case, PAW-AODV and AODV behave in virtually identical fashion. Subsequently, AODV shows better performance than PAW-AODV for some time but its performance eventually falls below that of PAW-AODV. In the initial stages, when nodes still have enough power, AODV carries more
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

packets because it has less operational overheads than PAW-AODV (fewer RREQ, RREP and RERRs are transmitted). As the nodes exhaust their battery power, AODV is at a disadvantage as it continues to route packets based on hop count even though the intermediate nodes may be on the verge of getting exhausted – once a node exhausts its battery power, not only are its own packets not carried but it is also no longer capable of routing packets of other node-pairs. In contrast, PAW-AODV is able to use the limited battery power more effectively since its calculations of route costs does take the nodes power level into account and penalizes routes which use nodes that have reached their “yellow” zones. Moreover, once a node reaches its “red” zone, it will refuse to route packets for others and will save its power for transmitting its own packets.

It may be noted that the performance of PAW-AODV begins to overtake that of AODV for $t \geq 5400s$ (Figure 3.8(a)). The difference in performance between them keeps increasing beyond the crossover point, with a difference of 32000 packets at $t=14000s$. In both Figures 3.8(a) and 3.8(b), it is observed that the plots for AODV, both for received packets and for remaining battery power, saturate quickly for $t \approx 5400s$ and more. This is because for $t \geq 5400s$, AODV has reached the stage where most nodes have exhausted their power so very few additional packets are actually carried to their respective destinations. This behavior is exhibited by PAW-AODV as well, but the threshold for the saturation behavior occurs much later and the system ends up carrying a much higher number of packets. The power conserving nature of PAW-AODV helps to distribute the power usage such that there are more nodes with enough power to transmit and route packets, so that the overall volume of packets carried (as indicated by Figure 3.8(a) at 14000s) is much higher. From Figure 3.8(c), it is observed that the average delay performance of PAW-AODV (0.193s) is somewhat poorer than that of
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

AODV (0.099s) because of the additional routing overheads. The difference in actual time is however not very noticeable (0.094s).

It should be noted that simulation experiments with various versions of PAW-AODV were conducted but not shown here. These include versions where nodes sent updates of their battery power status on their own at frequent intervals and ones where different values of $L$ were used. In the case where battery power updates were frequently made, the performance of PAW-AODV was significantly poorer than that shown in Figure 3.8. Apart from their transmission overhead and power consumption, these frequent updates also affected the performance of other nodes detrimentally because of the CSMA/CA MAC protocol. The performance gains expected from the node’s power updates were not enough to counter the performance loss caused by these additional overheads. The parameter $L$ indicated the number of outstanding path choices that the PAW-AODV protocol is able to take into account at intermediate nodes. Note that $L=1$ for AODV. Increasing $L$ also increases overheads as more RREQ and RREP packets will then get transmitted. On the other hand, while keeping $L$ low will restrict the number of path choices, $L=3$ is chosen as a compromise as the studies showed that $L=2$ is too small and that values higher than $L=3$ increased the RREQ/RREP overheads without a significant increase in the associated gains.

3.7.3 Mobility Scenario Results and Discussion

In the mobility test, AODV and PAW-AODV are both applied to the same simulated environment with the same traffic. However, the nodes are now mobile and the random waypoint model described in Section 3.7.1 is used. Figure 3.9 shows the graphical
interpretations of the same performance parameters (cumulative number of data packets received and remaining battery power over time, and delay statistics) for the mobile scenario as the ones shown earlier for the static case.

From Figure 3.9(a), it is observed that the crossover point, the point where the two line plots (AODV and PAW-AODV) intersect each other still at \(t=5400\)s, as for the stationary case. However, the cumulative number of packets received by each protocol is higher in the mobile scenario than in the stationary scenario considered earlier. At \(t=14000\)s, the
number of packets received using AODV is 134000 (as compared to 108000 in the stationary case) while that received using PAW-AODV is 150000 (as compared to 141000 in the stationary case). The difference in the number of packets received, between AODV and PAW-AODV, is 16000. The higher number of packets received in the mobility scenario may be explained by the fact that the mobility of nodes also encourages intermixing of the nodes in the network. This will take energy-depleted nodes away from heavy traffic routes and bring in fresh nodes with sufficient power to establish potential routes through them at different points in time. This ‘mixing’ effect can thus redistribute the power usage among all the nodes and aids in the efficient use of power overall. This is observed to be true for both AODV and PAW-AODV as this phenomenon is only dependent on the mobility of the nodes and not on the power conservation approach followed. The delay performance also shows the same kind of behavior trends for both AODV and PAW-AODV as in the case of the stationary scenario. As for the stationary case, it is noted that the average delay experienced by PAW-AODV is somewhat higher than that of AODV. The difference in actual time is however not very noticeable (0.097s).

The protocols are next simulated in the various mobile scenarios using the random waypoint model with the various pause times. AODV-\textit{P<value>} and PAW-AODV-\textit{P<value>} denote, respectively, the AODV and PAW-AODV systems with pause time set to \textit{value}, and \textit{value}=10, 30, 90 and 300 here. AODV-STAT and PAW-AODV-STAT implies that both protocols are simulated in a stationary condition but are randomly placed in the test area. Figure 3.10 shows the plots of the total number of data packets received as a function of the time of operation of the system for the AODV protocol (in
Figure 3.10(a)) and PAW-AODV protocol (in Figure 3.10(b)), where each protocol is evaluated with different pause time values set.

![Graph of Number of Packets Received against Time with the various Pause Times set](image)

Figure 3.10. Graphs of Number of Packets Received against Time with the various Pause Times set

Figure 3.10 shows that for both protocols, the number of packets received is higher when node mobility is higher (as represented by lower values of pause time). This is because mobility takes energy-depleting nodes away from heavy traffic routes and brings in nodes with sufficient power to establish potential routes through them at different points in time. This ‘mixing’ effect can thus redistribute the power usage.
among all the nodes and aids in the efficient use of power overall, leading to more packets being carried. PAW-AODV performs much better especially in highly mobile situations, as reflected by the huge improvement observed by comparing PAW-AODV-STAT and the rest of the PAW-AODV mobile plots in Figure 3.10(b).

3.7.4 System Performance with Limited Hop Count

In the versions of AODV and PAW-AODV considered earlier, the routing protocols can choose to set up paths of any length (as long as the routing rules and power constraint conditions are met). An interesting modification that will be investigated next was one where the routing paths selected are subjected to a hop count limit. In the case of PAW-AODV, this seems desirable from two points of view. Firstly, in situations where conserving power is important, the system should not only discourage paths where power is getting exhausted (which PAW-AODV does) but should also be more proactive and deny paths where the total power consumption would be high. This can be done by not allowing paths which have more hops than a pre-decided limit. Secondly, by preventing packets from taking inordinately long paths (with a large number of hops) the delay performance of the system will be improved as well. Preventing packets from traveling along low cost but high power consuming routes that involve a large number of hops will not only conserve battery power and force it to be more efficiently used but will also improve the overall delay performance. However, this hop count limit has to be judiciously chosen. Choosing too small a value will seriously impact the throughput as most traffic will be denied the routes required for transmission while choosing a hop count limit which is too high will make the modification ineffective.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

The hop count limit in the experiments conducted here will be set as the 90-percentile hop count value observed in the case of AODV (which is 4) in both stationary and mobile situations. The 90-percentile value indicates 90% of the data packets traverse a path of 4 hops or less to reach their destinations. To determine this 90-percentile value in a practical scenario, a node can track the hop counts of its self transmissions, and obtain the hop counts of other data transmissions via overhearing. The node can then compute the 90-percentile hop count value within a period or window of time and apply this to the next window. This online hop count tracking module is a future enhancement to be implemented in PAW-AODV. This hop count limit is applied to both AODV and PAW-AODV. AODV-ND4 and PAW-AODV-ND4 are denoted as AODV and PAW-AODV with hop count limit or NET_DIAMETER (as defined in [36]) set to 4, respectively. Both AODV-ND4 and PAW-AODV-ND4 are simulated in the stationary environment as well as the mobile environment.

Figure 3.11 shows the results obtained for the stationary simulation scenario. For comparison, Figure 3.11(a) shows the number of packets received for AODV and PAW-AODV, both for the case of ND=4 and the earlier case where no hop count limit was set. It may be noted that from Figure 3.11(a) that when \( t \) is small, i.e. \( t \leq 3500 \)s, all the four schemes perform in approximately the same fashion. The benefits of setting a hop count limit become evident only when the system has been operating for some time. In fact, when the hop count limit is set, alternative low cost paths longer than 4 hops will not be chosen. This will conserve more energy for transmitting more ‘short-haul’ packets that require 4 hops or less to reach their destinations. Thus, from Figure 3.11(a), PAW-AODV-ND4 performs consistently better than PAW-AODV throughout the entire
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

duration of the simulation run in terms of throughput. (AODV also benefits similarly from the hop count limit.)

![Graph of Number of Packets Received against Time](image1)

![Graph of Percentage of Power Remaining against Time](image2)

(a) Graph of Number of Packets Received against Time (b) Graph of Percentage of Power Remaining against Time

![Cumulative Delay Statistics at t=5400s](image3)

![Cumulative Delay Statistics at t=14000s](image4)

(c) Cumulative Delay Statistics at t=5400s (d) Cumulative Delay Statistics at t=14000s

Figure 3.11. Performance Comparison of AODV-ND4 and PAW-AODV-ND4 in a Stationary Scenario

Using the hop count limit to restrict path lengths also forces AODV–ND4 and PAW-AODV-ND4 to show similar battery power consumption behavior as can be seen in Figure 3.11(b). This trend is also shown by the delay performance of Figures 3.11(c) and (d) where the average delays of both AODV–ND4 and PAW-AODV-ND4 can be seen to be approximately similar. Comparing these results with those of Figures 3.8(c)
and (d), it can be seen that the delay performances of both protocols have improved because of the hop count limit but the improvement has been significantly higher in the PAW-AODV case. This has happened to the extent that the delay behavior of PAW-AODV-ND4 and AODV-ND4 are almost identical even in spite of the higher control overheads of the former.

Figure 3.12 shows the results obtained from the mobile scenario. These plots are very similar to those shown for the stationary case in Figure 3.11 and show the same overall trends. The cumulative number of packets received using PAW-AODV-ND4 (180000) is more than the corresponding number of packets received using AODV-ND4 (160000). The difference in the number of packets received with and without the hop count limit set is about the same for AODV (25000) and PAW-AODV (30000). Improvement for the number of packets received in the case of PAW-AODV-ND4 is once again higher than AODV-ND4 because of the “mixing” effects of mobility mentioned earlier. The power conservation effects of PAW-AODV-ND4 seem somewhat higher in Figure 3.12(b) than that observed for AODV-ND4. Comparing Figures 3.12(c) and (d) with Figures 3.9(c) and (d), the delay performances of both protocols are observed to improve when the hop count limit is used, more so for PAW-AODV than AODV. This agrees with the trends observed in the stationary case earlier.
Chapter 3 Power Aware Ad hoc On-demand Distance Vector Routing

(a) Graph of Number of Packets Received against Time
(b) Graph of Percentage of Power Remaining against Time

(c) Cumulative Delay Statistics at t=5400s
(d) Cumulative Delay Statistics at t=14000s

Figure 3.12. Performance Comparison of AODV-ND4 and PAW-AODV-ND4 in a Mobile Scenario

The protocols are next evaluated with different hop count limits imposed. Figure 3.13 shows the plots of the total number of data packets received as a function of the time of operation of the system for the AODV protocol (in Figure 3.13(a)) and PAW-AODV protocol (in Figure 3.13(b)), where different hop count limits are imposed. The default pause time of 30s is used here. AODV-ND<value> and PAW-AODV-ND<value> respectively denote the AODV and PAW-AODV systems with hop count limit set to value, and value=2 to 6 here. AODV and PAW-AODV (without any hop count limit set) are also included for comparison.
Figure 3.13 shows that if value=4, 5 or 6, the performances as reflected by the respective plots are similar to that without any hop count limit set. However, if value=3, which is the average hop count observed, with the sacrifice of not transmitting some longer-than-3-hop packets, the gain (from the energy conserved for not transmitting longer-than-3-hop packets) is more at the end of the simulation. Both AODV-ND3 and PAW-AODV-ND3 carry about 20000 packets each, as compared to the rest (excluding value=2) with less than 18000 packets each carried at t=14000s. Using value=2 is not recommended since the sacrifice is much more as shown from Figure 3.13, and the gain comes much later (at t>10000s) as compared to the case of value=3.
3.8 Conclusion

In this chapter, modifications to incorporate power aware features in the standard AODV protocol are proposed. This has been achieved in the new PAW-AODV protocol proposed. The performance of the PAW-AODV protocol was studied under various simulation scenarios and its performance is compared with that of standard AODV under both mobile and static conditions. In all cases, the power-aware features incorporated in PAW-AODV allow it to carry more packets overall, from source to destination, than standard AODV in a wireless ad hoc network. The delay performance of PAW-AODV is somewhat poorer than AODV. The difference in actual time is however not very noticeable. The delay performance of PAW-AODV can be made comparable to that of AODV by imposing a hop count constraint on the paths obtained by the routing approach. This also improves the throughput performance of both schemes since it improves the overall power efficiency by preventing the use of paths longer than the hop count limit. From the mobility test results, it is found that mobility can redistribute power usage, resulting in more packets being carried for both AODV and PAW-AODV protocols. The tests carried out to investigate the effect of setting different hop count limits show that setting an appropriate hop count limit for both AODV and PAW-AODV protocols helps conserve energy for carrying more packets at a later time.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

4.1 Introduction

In a commercial wireless ad hoc network, each node is an individual and may be reluctant to contribute its battery power resource. There are two concerns here that each node has, which are, whether its battery power resource is utilized fairly and efficiently, and whether the other nodes are also contributing their power resources to help the former carry its multi-hop packets to the respective destinations. To address the second concern, a Reward-based or Cost-Credit based system is implemented to enforce cooperation among nodes. A reward-based system is chosen due to the self-regulatory nature where a node can regulate its forwarding and self transmission traffic by monitoring its own credit level. This makes it a good cooperation-enforcement approach as mentioned in Section 2.3. To address the first concern, a forwarding rule based on a proposed forwarding strategy is implemented to ensure that a node is able to transmit as many self-generated packets as possible. This forwarding rule and its mode of operation in a simple system are discussed in Section 4.2. This rule is integrated in a proposed On-demand Cost-Credit Routing or ODCCR protocol and its Route Discovery and Management phases are discussed in detail in Section 4.3. Section 4.4 presents the ODCCR system’s performance in terms of network lifetime and throughput, and compares that to a typical AODV protocol and the Reward-based On-demand Routing or RBODR protocol when used in a power constrained static ad hoc network. Here RBODR is an on-demand routing protocol based on a reward system but where no forwarding rule is implemented. The performances of these three protocols in a power constrained mobile ad hoc network are also studied.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

The cost-credit based system is then examined from a centralized point of view. An optimization problem that aims to maximize the minimum battery power among all nodes when selecting routes for all the source nodes in the cost-credit based system is formulated in Section 4.5. A centralized max-min Power Credit Routing or mmPCR protocol is proposed to solve the optimization problem and perform centralized routing. This has also been included in Section 4.5. Its network lifetime and throughput performance results are discussed and compared in Section 4.6 with those of the ODCCR protocol. Conclusions regarding the proposed cost-credit based routing strategies are given in Section 4.7.

4.2 Cost-Credit based Forwarding

4.2.1 A Packet Forwarding Strategy in a Simple System Model

Consider the following forwarding strategy for a reward-based approach in a simple system. In a cost-credit game model, each ‘player’ or node will be given $C_i$ cost-credit units (CCUs) and $B$ battery units (BUs) initially. There are two ‘strategies’ that a node can play when a transit packet reaches it; either to forward or not to forward the packet. If the node forwards a transit packet, it will earn $c$ CCUs but loses $b$ BUs. If it does not forward then it will neither earn any CCUs nor lose any BUs. Here nodes will definitely need to forward and earn credits at some point in time because of these two conditions imposed - (i) $C_i$ is not large compared to $B$ and has only a nominal value to get transmissions started, (ii) if a node wants to transmit a self-generated packet, it needs to pay a cost of $c$, CCUs, on top of losing $b$ BUs.
In this system, time evolution is proceeding in a sequence of time slots of $\Delta s$ duration each, where $\Delta$ is very small. Let $c(t)$ and $b(t)$ be the respective CCUs and BUs available at a node at time slot $t$. There is a basic Transmission Rule (TR) here: “If $c(t) \geq c_r$ and $b(t) \geq b$ transmit a self-generated packet, else do not transmit.” In this simple model, $c < c_r$, and $c_r$, $c$ and $b$ are constants. Due to the self-regulatory nature, choosing a particular strategy (either to forward or not to forward) will directly affect a node’s ‘payout’ which reflects how many self-generated packets it can transmit. Consider a node which has already forwarded $N_F$ packets at time $t$. At time $t$, after forwarding $N_F$ packets, the total number of self-generated packets that it can potentially transmit is constrained either by the amount of cost-credit units it has $(c(t) + N_F c) / c_r$ or by the number of battery units it can allocate for self transmission $(B / b) - N_F$, whichever is less. Therefore, the payout function here is defined as:

$$w(t) = \min\left[\alpha_c, \alpha_b\right]$$  \hspace{1cm} (4.1)

The best strategy for a node is to always forward a packet for others if this does not decrease its overall payout function. In other words, a node should forward at time slot $t$ if its payout function will not decrease at time slot $(t + 1)$ as a result of this action, i.e. it should forward a packet at time slot $t$ if $w(t + 1) \geq w(t)$. A node may encounter three distinct situations at time slot $t$ where forwarding decision needs to be made -

(a) $\alpha_b < \alpha_c$ so $w(t) = \alpha_b$

(b) $\alpha_c + 1 > \alpha_b > \alpha_c$ so $w(t) = \alpha_c$

(c) $\alpha_b \geq \alpha_c + 1$ so $w(t) = \alpha_c$
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

If it forwards under situations (a) and (b), this will result in \( w(t + 1) < w(t) \). However, if it forwards under situation (c), then \( w(t + 1) \geq w(t) \). Therefore, a node should forward only if (c) holds. This may be restated as:

\[
\frac{B}{b} - N_F \geq \frac{C_I + N_F c}{c_r} + 1 \quad (4.2)
\]

Let \( N_T \) be the number of self-generated packets transmitted by the node until time slot \( t \). This will give \( c(t) = C_I + N_F c - N_T c_r \) and \( b(t) = B - (N_F + N_I)b \), so substituting into (4.2) will result in:

\[
\frac{c(t)}{c_r} \leq \frac{b(t)}{b} - 1 \quad (4.3)
\]

In this simple system, if a node knows its current cost-credit and battery levels, by evaluating (4.3), it can make a good forwarding decision. Based on this packet forwarding strategy, a forwarding rule (FR) formulated for this system is: “If

\[
\left[ \frac{c(t)}{c_r} \leq \frac{b(t)}{b} - 1 \right] \text{forward a packet, else do not forward.}
\]

4.2.2 Operation in a Simple System

When in operation, the initial condition \( \left( \frac{C_I}{c_r} \leq \frac{B}{b} - 1 \right) \) must be satisfied in each node as otherwise, the nodes will never participate in the forwarding game, and the ad hoc network will be unable to carry multi-hop transmissions. Given that the above initial condition is satisfied, if a node always follows the forwarding rule FR, it will only need to forward till \( N_F = N_{F,max} \), as then it would have earned enough credits to transmit the
maximum number of self generated packets \( (N_{T,\text{max}}) \) allowed by its battery power resource.

\[
N_{F,\text{max}} = \left( \frac{(B-b)c_r-C_gb}{b(c+c_r)} \right) + 1 \tag{4.4}
\]

\[
N_{T,\text{max}} = \min \left[ \left( \frac{C_r + N_{F,\text{max}}C}{c_r} \right), \left( \frac{B}{b} - N_{F,\text{max}} \right) \right] \tag{4.5}
\]

By adopting the proposed forwarding strategy, the payout function value of a node will never decrease, but will eventually increase to the maximum value indicated in (4.5). This value is located within the crossover region shown in Figure 4.1, which is a plot of the payout function \( w(t) \) against \( N_F \). If every other node adopts this strategy, an individual can either adopt this strategy as well, or deviate by trying the following tactics:

(a) Continue forwarding one or more packets even if \( N_F \geq N_{F,\text{max}} \).

(b) Stop forwarding completely or not forward packets for some time when \( N_F < N_{F,\text{max}} \).

Trying tactic (a) will cause the payout function to be moved to region 2 and this would decrease the payout value. On the other hand, when a node uses tactic (b) and stops forwarding completely when its payout function is still within region 1, it will be depriving itself from getting a higher payout value should it continue to forward till the crossover region. If a node using tactic (b) chooses to stop forwarding for some time when its payout function is still within region 1, it may not be able to go over to the crossover region when it decides to forward later on. This is because at that moment, other nodes may not need to transmit packets via this node, or they may have completed their own transmissions. In all these cases, the deviating node will not be able to get
more than the maximum payout $N_{t,\text{max}}$. Thus, adopting the proposed forwarding strategy is the best option for the node. Since every node finds this forwarding strategy to be its best strategy and finds no reason to deviate, this situation is similar to a Nash Equilibrium. The forwarding rule encourages all nodes to be cooperative in making a multi-hop ad hoc network operational, and it gives them a very good reason to cooperate, i.e., maximum self transmission can be achieved by always cooperating.

![Figure 4.1. Graph of payout function $w(t)$ against $N_F$](image)

4.2.3 Operation in a Real System

The forwarding rule FR derived in Section 4.2.1 is similar to Rule 1 in [72]. However, the approach here is convenient to derive approximate heuristic algorithms which may be practical to implement in an on-demand routing protocol to be discussed later. In a real system, $c_r$ in FR cannot be a constant. This is because $c_r$ CCUs paid per forwarding
packet will be distributed among all the forwarding nodes and destination on the route selected. Since each forwarding node earns \( c \) CCUs per packet forwarded, \( c_r \) will be a hop-length dependent value. In the real system, the value of this \( c_r \) is set as a moving window average value, which is computed as an average of all the route costs of the self transmissions observed till the moment whenever FR needs to be evaluated. This implies that for a node that has transmitted its \((i+1)^{th}\) self-generated packet with cost \( c_r \), the average route cost observed thus far, \( c_{r,i+1} \) as updated in (4.6), will be used in place of \( c_r \) in FR. Using a moving window average method to compute the average route cost has the advantage of keeping the average route cost updated based on transmissions observed thus far.

\[
c_{r,i+1} = (1 - \alpha)c_{r,i} + \alpha c_r
\]  

(4.6)

\( \alpha = 0.01 \) is used in the simulations here though other values of \( \alpha \) have been tested to study its effect on system transients. Note that (4.6) cannot be applied in a situation where a node has yet to transmit any self generated packet and it needs to check FR for making a forwarding decision. For this case an initial \( c_r = c_{r,0} \) is defined. \( c_{r,0} \approx 2.5c \) is chosen as an approximation based on geometrical arguments for an infinite area with nodes uniformly and randomly distributed. This is further discussed in Appendix A. However, simulation results are found to be not very sensitive to the initial choice of \( c_{r,0} \) and \( c_{r,0} \approx 2.5c \) can be appropriately applied in the 2-dimensional networks to be studied later.
4.3 On-demand Cost-Credit Routing

4.3.1 Route Discovery

An on-demand routing protocol which has the forwarding rule (FR) integrated into it has been designed to operate using a reward-based or cost-credit system. On-demand routing protocol is chosen for development as it can easily handle both static as well as mobile situations. Table-driven routing protocols require all nodes to exchange topology information periodically, so that each node can maintain routes to all other nodes in its route table at all times. These protocols will not work well in mobile situations, where a large amount of control packets will be generated for each node to keep track of the positions of all the other nodes whenever they move. The route table may also not get updated in time. An on-demand routing protocol may be a better option since it requires a node to initiate a route discovery process to find a route only when it needs one to connect to a destination. This gives it a more updated route table and less overhead is involved. The reward-based on-demand protocol proposed here is referred to as On-Demand Cost-Credit Routing (ODCCR) protocol. ODCCR is modeled after AODV [36] because out of the two commonly used on-demand routing protocols (AODV and DSR), AODV performs better than DSR in scenarios with heavier network load and/or higher node mobility and is also more scalable than DSR [37]. A secured cost-credit module is assumed to be in place to handle proper cost deduction for transmitting self-generated packets and for credit accumulation as a result of forwarding. The system security measures required will not be elaborated further as the emphasis here is on the design of an on-demand routing protocol to operate using FR and the cost-credit system.
Figure 4.2 gives an overview of the route discovery process of ODCCR. This process starts when a source needs to transmit to a destination for which it does not have an entry in its route table. A RREQ packet, similar in structure to a standard AODV RREQ packet, will be initialized and constructed. The only difference is that in the RREQ here, route_cost field is used instead of hop_count field. As the RREQ propagates from the source to the destination, each node along the route (excluding the destination) will add its node cost into the route_cost field so as to build up the reverse route cost to the source. Therefore, the source initializes the route_cost field of RREQ with its own node cost before it broadcast this control packet. Note that given that $c_i$ is the node cost of node $i$, route cost $c_r$ is the summation of the node costs of all the forwarding nodes and destination on route $r$, which is

$$c_r = \sum_{i \in r} c_i$$ \hfill (4.7)

In the cost-credit model mentioned, $c_r$ is the CCUs required to be paid for self transmission. Since $c$ CCUs are earned for forwarding, this is assumed to be the fixed node cost $c_i$ of each node $i$. The destination needs to be paid $c$ CCUs so that no self transmission comes free, even for one-hop self transmission. However, the destination will not earn these CCUs as it is not a forwarder. The route cost here is therefore $c_r = nc$, where $n$ is the number of hops in the selected route. Even though the node cost is assumed to be a constant, in a real system it can be a variable dependent on parameters like the node’s current battery level and/or its cost-credit level.

If a forwarding node encounters a RREQ, it will first check to see how many times it has seen the same RREQ within a broadcasting interval. When a source broadcasts a RREQ to find a route to a destination, it will set a unique broadcast_id field value in this
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

RREQ. This source will send another RREQ with a different broadcast_id field value if it does not get a reply within a waiting period (similar to the standard AODV RREQ broadcasting interval). The forwarding node will drop the RREQ with the same broadcast_id and source_address if it has seen this same RREQ at least L times within a waiting period. Otherwise, it will proceed to check the reverse route to source in its route table. Setting an appropriate value for L is important. Setting L>1 allows more routes of possibly lower cost to be discovered, but it should not be set too high a value as that will significantly increase the traffic volume of control packets like RREQ. L=3 is found to be a suitable choice in the simulation studies. When the node proceeds to check the reverse route in its route table, it will either place a route entry pointing to the source if it does not have such an entry, or it will update the route entry if the RREQ carries an updated reverse route to source with a higher source_sequence_no or one with a lower route cost. If no entry or updating is made, the RREQ will be dropped. The route table here is similar to the standard AODV route table, except that there is a route_cost column instead of a hop_count column, so the route cost is entered or updated accordingly for a route entry. The node will next proceed to check the forwarding rule (FR). Should the FR require the node not to forward, it will drop the RREQ. Otherwise, it will next check its route table for an updated forward route pointing to the required destination. If this node has an updated forward route with a destination_sequence_no greater than or equal to the destination_sequence_no field of the RREQ, it will initialize and construct an RREP packet. The RREP here is similar in structure to the standard AODV RREP packet, except that again, route_cost field is used instead of hop_count field. The node will add the forward route cost from its route table and its own node cost into this route_cost field of the RREP before forwarding it along the reverse route towards the source. If the node does not have an updated forward
route, it will just add its node cost into the route_cost field of the RREQ before re-broadcasting this control packet.

The RREQ will be broadcast as above until it is received by the destination node. First, the destination node will drop the RREQ with the same broadcast_id and source_address if it has seen this same RREQ at least L times within a broadcasting interval. Else, it will proceed to check the reverse route pointing to the source in its route table. The destination will add the reverse route and associated route cost to its route table if there is no such route, or update the reverse route and route cost if the RREQ carries an updated reverse route or one with lower route cost. If no addition or updating of reverse route is performed, the RREQ will be dropped. Otherwise, the destination node will construct a RREP and initializes the route_cost field of the RREP with its node cost before transmitting the RREP along the reverse route towards the source. Whenever an intermediate node intercepts a RREP packet, it will first check its route table for any addition or updating that needs to be done. This intermediate node will add the forward route pointing to the destination and the associated route cost if no such route can be found in its route table. Otherwise, it will update the forward route if the RREP carries an updated forward route with a higher destination_sequence_no or one with a lower route cost. If no addition or updating is performed, the RREP is dropped. Otherwise, the intermediate node will add its node cost into route_cost field of the RREP before forwarding this along the reverse route towards the source. Based on the information carried in the all the RREP packets, the source will select the least cost route. Before it can transmit, it will need to check the Transmission Rule (TR). If TR can be satisfied, the source will be allowed to transmit the packet. Otherwise, the packet will remain in the buffer until enough credits have been earned from forwarding.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Type of Node

Source node?

Source has data to send

Include node cost in route_cost field and send RREQ

Intermediate node?

Receive RREQ?

Were at least L RREQs seen?

Reverse route updated or of lower cost in table?

Process RREP

Forward route updated or of lower cost in table?

Include node cost in route_cost field and send RREP towards source

Y

Drop RREQ

Y

Drop RREQ

Y

Source waits for enough credits earned to send again

N

TR satisfied?

Receive RREP?

Y

Include node cost in route_cost field and send RREQ

Y

Drop RREQ

Y

Record in table, include node cost in route_cost field and send RREP towards source

N

Update reverse route in table

Y

Include route cost to destination and node cost in route_cost field and send RREP.

N

Update forward route available?

Y

Drop RREQ

Figure 4.2. Route Discovery Process of ODCCR
4.3.2 Route Management

Upon obtaining the lowest cost route, and having satisfied the TR, a source will deduct a cost (as discovered during route discovery) from its credit pool before transmitting a self-generated data packet. When a forwarding node receives the data packet, an amount $c$ will be added to its credit pool. (The destination node does not earn any credits for the received packet.) During data transmission, a time may come when a forwarding node finds that its FR is violated. When this happens, the node will still continue to forward the packets that are already in its transit buffer. It will also broadcast a route error (RERR) packet upstream to deny any future forwarding. When an upstream forwarding node receives the RERR, it will drop all data packets that need to be forwarded by the RERR initiator. These upstream forwarding nodes will also broadcast the RERR upstream towards the source node. The RERR will be transmitted in this manner till it reaches the source node. Upon receiving the RERR, the source node will then initiate a route discovery process again if so required. The RERR initiator can resume forwarding again at any point in time when it finds that its FR is not violated anymore, as it needs to earn credits again for future self transmission. All it needs to do is to start replying to any route requests for forwarding to proceed via it. In the event of a link breakage due to node mobility, similar route error propagation and handling processes will be triggered. In this case, RERR will be broadcast upstream to those nodes affected by that broken link.
4.4 Performance Studies

4.4.1 Static Network Results and Discussions

QUALNET version 3.7 developed by Scalable Network Technologies Inc [79] is used as the network simulator here to evaluate the throughput performances of AODV, RBODR (Reward-based On-demand Routing) and ODCCR in a power constrained static ad hoc network. RBODR is similar to ODCCR, except that FR is not implemented and is therefore not applied for (a) processing RREQ during route discovery and (b) during data transmission or forwarding. However, TR is still used in RBODR and nodes do need to earn credits to transmit self-generated packets. The absence of FR in RBODR means that packet forwarding and credit earning behaviors of nodes are not regulated. Thus, the purpose here is to evaluate ODCCR, a reward-based system with TR and FR implemented, against a non reward-based system (AODV) and a reward-based system with TR implemented only (RBODR). In all these systems, forwarding and transmission ceases at nodes when they run out of battery power. The performance studies will focus on the network lifetime and the self-generated packets that nodes can transmit/receive (i.e. the throughput) within the operational network lifetime.

The network topology used is a square network where 720 nodes are randomly placed. Since long hop-lengths would be impractical in an ad hoc network, the simulated scenarios require that a source can only send to destinations which are not more than three radio-ranges away. In every 4-minute interval, each node will make a 2.5-second connection and generate one 64-byte packet every 0.5 second. Each node has initial battery power $B=7000$ and $c=1$, $b=1$ is assumed. Two values of $C_I$ are set here: $C_I = 0.6B$ and $C_I = 1.5B$. Results are obtained only from the nodes in the central region of the
network so as to ignore end-effects that will affect the behavior of the nodes near the boundaries of the overall area.

There are two definitions of *Network Lifetime* \([49]\), which are (a) time \(T_F\) until the first node runs out of power, or (b) time \(T_K\) until \(K\%\) of the nodes run out of power. Both \(T_F\) (indicated by the circles at the start of respective line graphs) and \(T_K\) (respective line graphs plotted for \(K \leq 20\)) are shown in Figure 4.3(a) for \(C_I = 0.6B\) and Figure 4.3(b) for \(C_I = 1.5B\). It is evident that for both definitions (a) and (b), ODCCR’s lifetime is much longer than those for RBODR and AODV systems. This implies that using FR is important in conserving energy for self transmissions, which differentiates ODCCR from a simple reward-based augmentation of AODV (i.e. as in RBODR). It is interesting to observe that while lifetime values increase relatively rapidly when \(K\) is small (or for \(T_F\)), the increase is much slower for larger \(K\). In this latter region (i.e. for \(K\) beyond this threshold), a longer network lifetime is observed because the nodes are unable to transmit multi-hop packets since they have insufficient credits, or because some of the nodes have already lost power and cannot act as intermediate nodes any more. This should not be considered a good operating range for the system as very little additional traffic is actually carried once the system enters this region; here, even though the nodes retain power (because they are unable to transmit) and the system shows longer lifetimes, this extra life does not actually contribute much additional throughput.

Throughput performance is shown in Figure 4.4 and 4.5 through plots of average values of \(N_T\) and \(N_R\) versus \(K\) (as well as for \(T_F\)). Here, \(N_T\) is the average number of packets actually transmitted by a node and \(N_R\) is the average number of these received at their respective destinations (all averaged over the central nodes for the network lifetime, \(T_F\) or \(T_K\)). Note that packets may be dropped by intermediate nodes when paths break or
nodes lose power. As in Figure 4.3, the proposed ODCCR approach performs much better than the other two schemes. (RBODR appears to be slightly inferior to AODV for these performance measures.)

Figure 4.3. Comparison of Network Lifetimes

(a) $C_I = 0.6B$

(b) $C_I = 1.5B$
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Figure 4.4. Comparison of Number of Packets Transmitted

(a) $C_I = 0.6B$

(b) $C_I = 1.5B$
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Figure 4.5. Comparison of Number of Packets Carried

(a) $C_I = 0.6B$

(b) $C_I = 1.5B$
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Most literatures [82]-[84] focusing on network lifetime performance define network lifetime as the time taken for the first node to run out of battery power and get deactivated. In view of this, we have conducted more experiments using this definition of network lifetime. The nodes are now randomly and uniformly distributed in a very large area of size $13r \times 13r$, where $r$ is the radio range ($r=418m$). Again, simulations results will be obtained only from the nodes in the central region, within an area $3r \times 3r$. Table 4.1 summarizes the values assumed by the various parameters used in these simulation studies.

Table 4.1. Values of the Parameters used

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of self-generated connections per second per node, $N_m$</td>
<td>1/300 to 1/1800</td>
</tr>
<tr>
<td>Number of packets generated per connection, $r_m$</td>
<td>10</td>
</tr>
<tr>
<td>Packet size</td>
<td>64 bytes</td>
</tr>
<tr>
<td>Inter-arrival time</td>
<td>0.5s</td>
</tr>
<tr>
<td>Battery power unit lost per packet, $b$</td>
<td>1</td>
</tr>
<tr>
<td>Credit unit gained per forwarded packet, $c$</td>
<td>1</td>
</tr>
<tr>
<td>Initial battery power, $B$</td>
<td>$300b$ and $150b$</td>
</tr>
<tr>
<td>Radio range, $r$</td>
<td>$418m$</td>
</tr>
</tbody>
</table>

To see the effect of having more nodes to handle routing, the node density $\rho$ is varied from 5 nodes per $r^2$ m$^2$ to 30 nodes per $r^2$ m$^2$ at $C_B/B = 1$ where the nodes are randomly placed in each simulation run. However, in order to have the same level of medium access contention for each density distribution, $\rho N_m$ is kept constant, i.e. at $\rho N_m=(1/60)$ connections per $r^2$ m$^2$ per second. (This is done so that the results obtained are not greatly affected by changes in the medium access contention effects that may arise due to changes in the node density distribution). Here $N_m$ is the number of self-generated connections per second per node. Note that $N_m$ is constant in a simulation run (not random) and will vary from 1/300 connections per second per node ($\rho=5$) to 1/1800 connections per second per node ($\rho=30$) for simulations done with different $\rho$. During
each connection, a fixed number of \( r_m \) packets (of 64 bytes each) are generated at an inter-arrival time of 0.5 seconds. The network lifetime results obtained through the simulations are shown in Figure 4.6(a). Note that Figure 4.6(a) actually plots the Normalized Network Lifetime or \( \frac{\text{Network Lifetime}}{\text{(Connection Inter-arrival Time per node)}} \) as a function of the node density \( \rho \) for constant \( \rho N_m=(1/60) \). The network lifetime here is normalized with respect to the connection inter-arrival time per node, i.e. \( 1/N_m \).

The packet load and throughput behavior of the simulated system as a function of the node density \( \rho \) (for constant \( \rho N_m \)) are shown in Figures 4.7(a) and 4.8(a) respectively for nodes in the central \( 3r \times 3r \) region. Figure 4.7(a) shows the average number of self-generated packets transmitted during the network lifetime while Figure 4.8(a) gives the average number of self-generated packets that reached their respective destination. It may be noted that for each point shown in the simulation results of Figures 4.6, 4.7 and 4.8, results obtained from multiple simulation runs were averaged. During each run, different source-destination pairs were randomly selected for each connection. The simulations imposed a hop-limit of 3, i.e. routes with more than three radio hops were discarded and packets which could not be carried over routes of three or fewer radio hops were discarded and lost.

As shown in Figures 4.6(a), 4.7(a) and 4.8(a), ODCCR shows longer network lifetimes and higher throughputs than both AODV and the RBODR systems, regardless of the node density. For all protocols, the normalized network lifetime decreases as node density \( \rho \) increases from 5 nodes per \( r^2 \) m\(^2\) because it becomes easier to find forwarding nodes, as shown in Figure 4.6(a). This leads to less data packets dropping and more successful connections being made. With more data packets transmitted and forwarded, the normalized network lifetime becomes shorter. Beyond \( \rho=20 \) nodes per \( r^2 \) m\(^2\), the
network lifetime does not change much. In the case of AODV, there is no restriction on the number of packets a node can forward, as compared to ODCCR which has a FR. This FR conserves energy for each node and prolongs the node lifetime so that the node can transmit more self-generated packets and forwards fewer packets. This implies that by the time a node (in the case of AODV) deactivates and ends its life, the maximum number of self-generated packets transmitted is determined by how many packets it has forwarded. In the case of ODCCR, the maximum number of self-generated packets to be transmitted by a node is predetermined by FR and must happen before a node ends its life. When the node density increases, the average number of packets transmitted and the corresponding number of packets carried successfully to the destinations per node decrease in the case of AODV as shown in Figures 4.7(a) and 4.8(a), due to the decreasing normalized network lifetime. The performance results of AODV and RBODR are quite close since each node in the case of RBODR has enough credits (with $C/B = 1.0$) such that RBODR operates like AODV. However, FR in the case of ODCCR dictates the maximum number of packets to forward. Near the end of the network lifetime, a substantial number of central nodes will have violated their FRs and would only be transmitting self-generated packets. When node density increases, nodes need less time to successfully transmit self-generated packets as the probability of finding forwarding nodes increases. This implies that an average number of self-generated packets that is successfully transmitted within a certain network lifetime now can be achieved within a shorter network lifetime when node density increases. Therefore, although the network lifetime decreases as node density increases, the average number of self-generated packets that is successfully transmitted does not vary much. Therefore, the average number of packets successfully carried to the destinations does not vary as much as that in the case of AODV and RBODR.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

(a) Normalized Network Lifetime against Node Density at $C/B = 1$. 
$\rho N_m = (1/60)$ connections per $r^2$ m$^2$ per second

(b) Network Lifetime against $C/B$ at Node Density = 10 nodes per $r^2$ m$^2$

Figure 4.6. Network Lifetime Performance Comparison
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

(a) Average Number of Packets Transmitted against Node Density at $C/B = 1$

(b) Average Number of Packets Transmitted against $C/B$ at Node Density = 10 nodes per $r^2$ m$^2$

Figure 4.7. Average Number of Data Packets Transmitted per Node within the Operational Network Lifetime
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Figure 4.8. Average Number of Data Packets Carried per Node within the Operational Network Lifetime

(a) Average Number of Packets Carried against Node Density at $C_i/B = 1$

(b) Average Number of Packets Carried against $C_i/B$ at Node Density = 10 nodes per $r^2 m^2$
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

The behaviors of reward-based or cost-credit routing protocols like RBODR and ODCCR, when assigned different initial amount of credits, are also investigated. For this, the initial amount of credits is varied with respect to the initial battery level for a fixed value of $B$. $C/B$ is varied from 0.25, where nodes need to try very hard to earn credits for transmission, to 1.5 where nodes need to spend less time earning credits while keeping the node density constant at 10 nodes per $r^2$ m$^2$. Figures 4.6(b), 4.7(b) and 4.8(b) show the network lifetime, the average number of self-generated packets transmitted within the operational network lifetime and the average number of self-generated packets that reached their respective destination respectively for the three protocols when $C/B$ is varied. As the performance of AODV is unaffected by the variation of $C/B$, its results are indicated in a text box in the respective figures. For the case where the initial battery power $B=300b$, the results for each protocols are represented by $<protocol>(B=300b)$. At $C/B = 0.25$, with less credits given in the beginning, RBODR and ODCCR will operate in a similar manner, where there will be a point in time within the network lifetime, when nodes will run out of credits in their reserves. Thereafter, the number of self generated packets that can be transmitted is constrained by the amount of credits earned. Despite the small initial amount of credits given, RBODR and ODCCR perform as well as AODV in terms of the average number of data packets transmitted and carried within the operational network lifetime as shown in Figures 4.7(b) and 4.8(b). As $C/B$ increases, RBODR operates in a manner similar to AODV, where nodes will not run out of credits in their reserves and therefore can transmit all self-generated packets within the network lifetime. Figures 4.6(b), 4.7(b) and 4.8(b) show that RBODR exhibits an almost constant trend-line similar to the constant results of AODV when $C/B \geq 0.4$. In the case of ODCCR, it is desirable to operate when $C/B \geq 0.4$ as there will be no constraint on self transmission due to the
unavailability of credits. At the same time, increasing $C/B$ will decrease the number of packets a node needs to forward due to the FR implementation. Therefore, as observed from Figures 4.6(b), 4.7(b) and 4.8(b) that as $C/B$ increases, the network lifetime and the average number of data packets transmitted and carried to the respective destinations increase as well. If the initial battery of each node is decreased (setting $B=150b$), the three protocols still behave in the same manner as discussed earlier. The only difference is that the trend-lines as represented by $<protocol>(B=150b)$ for each respective protocol are lower now that each node is given a lower initial battery power.

Although increasing $C/B$ will decrease the number of packets a node needs to forward in the case of ODCCR, this will also decrease the number of multi-hop packets it can transmit, since other nodes are as reluctant to forward as well. Figure 4.9 shows the proportion of total data packets transmitted that belongs to forwarding packets within the operational network lifetime when $C/B$ varies in the case of ODCCR. When $C/B > 1.0$, the proportion of total data packets transmitted that are packets forwarded decreases sharply. This corresponds to a ‘knee’ and steeply increasing network lifetime in the case of ODCCR when $C/B$ increases beyond 1.0. From this point onwards, the proportion of one-hop self generated data packets increases sharply as well. Operating beyond this point is not very meaningful in the context of a multi-hop ad hoc network as the bulk of the data packets transmitted in this case are one-hop data packets. Thus, with an appropriate value of $C/B$ set (between 0.4 and 1.0) for ODCCR, the network lifetime, average number of data packets transmitted and carried within the operational network lifetime are substantially higher as compared to the case of the other two protocols, while the multi-hop transmissions are not significantly affected.
From all the simulations conducted, it is observed that the average end-to-end delays pertaining to ODCCR and RBODR (within the range of 1s to 4s) are higher as compared to the average end-to-end delay incurred in the case of AODV (within the range of 0.02s to 0.05s). This is because for both ODCCR and RBODR, source nodes will explore more paths during RREQ propagation and they will wait for a period of time for RREP gathering before they select the least cost path to transmit data packets. Also, at any point in time, if a source node does not have any credits, it will need to delay its data transmission until it has forwarded some data packets and earn enough credits to ‘pay’ for its own transmission. Nevertheless, despite the fact that both ODCCR and RBODR perform equally well in terms of delay performance, ODCCR outperforms RBODR in terms of lifetime and throughput performance because the former implemented FR to regulate the number of data packets to forward so as to increase the number of self-transmitted data packets.

Figure 4.9. Proportion of total data packets transmitted that are packets forwarded for others when \( C/B \) varies in the case of ODCCR
4.4.2 Mobile Network Results and Discussions

The performance of ODCCR is also studied in a mobile scenario (using QUALNET). Its performance is compared with that of AODV and RBODR under the same conditions. For these simulations, 180 nodes are uniformly placed in a 6D X 6D square area. (The mobile scenarios here are similar to those in [38]). In each mobile scenario, each node moves in a random direction using the random waypoint model with a speed randomly chosen (uniformly) within the range of 0-20 m/s. In between movements, a node will remain stationary for a period indicated by a specified pause time. A different pause time is selected for each mobile scenario so as to evaluate the performance of the three protocols when the pause time varies from 50s (highly mobile) to 500s (least mobile). The movements of all nodes are restricted to the square region. Forty source-destination pairs are chosen randomly within each interval of 60 seconds and the number of data packets initiated for each connection is within the range [5, 15] at a rate of two 64-byte packets per second. Given that $b=c=1$ and each node is assigned $300b$ BU$\lceil$s initially. $C/B=0.8$ is chosen for ODCCR, which is within the suggested range (between 0.4 and 1.0) as discussed in Section 4.4.1.

Figures 4.10(a), 4.10(b) and 4.10(c) show the network lifetime, the average number of data packets transmitted and the average number of data packets carried to the destinations within the operational network lifetime, respectively; these are plotted against the pause time for all the three protocols. Each point shown in the simulation results of Figure 4.10 is the average result obtained from multiple simulation runs. During each run, different source-destination pairs were randomly selected for each connection. As before, the simulations imposed a hop-limit of 3. Figure 4.10 shows that
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

*ODCCR* performs better than the other two protocols, as its route management module can handle mobile situations as well. The announcement of link breakage and usage of updated routes as established on demand in the case of *ODCCR* is effective in handling mobile situations. Together with its FR, this extends the network lifetime and also results in significantly more data packets being transmitted and carried. The performance degrades somewhat when node mobility is very high because of frequent link breakages. (This is to be expected for all the protocols.) Nevertheless, even at a pause time of 50s (i.e. highest mobility), *ODCCR* is still able to perform as well as the other two protocols. These plots show that even though *RBODR* can also handle mobile situations with its route maintenance module, it will not perform as well as *ODCCR* since it does not use the FR to control forwarding. An observable difference between *RBODR* and *AODV* is noticed in the mobile scenario, as compared to the previous static situations when the performances of the two were close to each other. Node movement causes forwarding traffic to vary substantially at times and in the case of *RBODR*, the nodes need more time to earn credits. This prolongs the network lifetime and increases the average number of data packets transmitted and carried within the operational network lifetime.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Figure 4.10. Performance Comparison in Mobile Scenarios

(a) Network Lifetime against Pause Time

(b) Average Number of Packets Transmitted against Pause Time
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

4.5 Centralized Power Credit Routing

4.5.1 General Network Lifetime Formulation

Let $B$ be the initial battery power of each node $i \in N$ set of nodes. Transmitting a packet loses $b$ BUs. The energy consumed for reception is very small as compared to the energy required for transmission and thus the battery power lost for receiving data packets is ignored [31], [77], [82]. This is applicable in our scenarios as nodes are randomly placed within the radio range of a source node and the likelihood of two nodes being very close is small. A large separation will result in a small negligible amount of reception energy as compared to the transmission energy required. It may be recalled that the network lifetime is the time till the first node has run out of battery power. For a power constrained network, the network lifetime maximization problem is formulated as
follows assuming that each node has at least one route to every other node and that all routes are available and unchanged for the static scenario considered.

\[
\max T \\
\text{subject to: } Q_i(T) \geq 0, R_i(T) \geq 0, \forall i \in N \tag{4.9}
\]

\[
(Q_i(T) + R_i(T))b \leq B \quad \forall i \in N \tag{4.10}
\]

\(Q_i(T)\) and \(R_i(T)\) are respectively, the actual number of packets forwarded and actual number of self-generated packets transmitted by node \(i\) till time \(T\), which is the network lifetime. (4.10) indicates the power constraint faced by each node. In the above formulation, \(R_i(T) = Q_i(T) = 0 \forall i \in N\) cannot be a solution. This is because \(R_i(T) \forall i \in N\) are given values that must be satisfied if possible. Unless a route cannot be found for a source-destination pair, all the data packets generated by a node should be transmitted if possible. \(Q_i(T) \forall i \in N\) are variables of which the respective values have to be found by solving the optimization problem using a routing strategy to be proposed later on. The proposed routing strategy will select a route that needs some intermediate nodes to forward for a particular source-destination pair. Different nodes will participate in forwarding at different times. Thus, \(Q_i(T) = 0 \forall i \in N\) cannot be a solution.

Consider the time evolution of the network as proceeding in a sequence of rounds or intervals, where \(t(k)\) indicates the duration of the \(k\)-th interval. Given that \(K\) indicates the number of intervals that has elapsed till the first node runs out of power, this results in:

\[
T = \sum_{k=1}^{K} t(k) \tag{4.11}
\]
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

The problem can be reformulated as maximizing the number of intervals that has elapsed till the first node runs out of power, as shown below. In this formulation, each interval is of equal duration. The associated power constraints are then indicated from (4.13) to (4.16), where \( r_i(k) \) and \( q_i(k) \) are the number of self-generated and forwarding packets transmitted at node \( i \) in the \( k \)-th interval. \( r_i(k) \) and \( q_i(k) \) may vary from interval to interval. \( Q_i(K) \) and \( R_i(K) \) are the total number of self-generated and forwarding packets transmitted at node \( i \) till the end of the \( K \)-th interval.

\[
\max K
\]

subject to:

\[
\begin{align*}
0 & \leq r_i(k) \leq 0 & & \forall i \in N, k > 0
\end{align*}
\]

\[
R_i(K) = \sum_{k=1}^{K} r_i(k) & \forall i \in N
\]

\[
Q_i(K) = \sum_{k=1}^{K} q_i(k) & \forall i \in N
\]

\[
(R_i(K) + Q_i(K))b \leq B & \forall i \in N
\]

For a cost-credit system, not all packets \( s_i(k) \) generated at the beginning on the \( k \)-th interval can be transmitted within the same interval. The number of packets \( r_i(k) \) that can be transmitted within the \( k \)-th interval is dependent on credit level at the beginning of the interval and the route cost per packet to be transmitted. The number of packets that cannot be transmitted will be deposited in the buffer and will remain there until the node has earned enough credits to transmit them. The number of packets that has been brought forward from the \( k^* \)-th interval and still exists in the buffer at the beginning of the \( k \)-th interval is represented as \( x_i(k^*, k) \), where \( k^* < k \). Therefore, the total number of packets \( X_i(k) \) in the buffer of node \( i \) at the beginning of the \( k \)-th interval will be given
by (4.17). This implies that the maximum amount of packets that a node should transmit within an interval is given by (4.18).

\[
X_i(k) = \begin{cases} 
0 & k = 1 \\
\sum_{k=1}^{k-1} X_i(k^*, k) & k > 1 \\
\forall i \in N 
\end{cases}
\]  

(4.17)

\[
r_i(k) \leq s_i(k) + X_i(k) \forall i \in N, k > 0
\]  

(4.18)

The lifetime maximization problem (4.12) to (4.16) in the cost-credit system is to be solved as follows. In each \( k \)-th interval, if a source node has more than one route to a particular destination, it will only consider the routes that allow maximum transmission based on its current credit level and the route cost of each available route; otherwise, the only route available is chosen. This gives \( r_i(k) \forall i \in N \) as a fixed requirement for the \( k \)-th interval. The only variables here are \( q_i(k) \forall i \in N \) so a set of routes has to be found such that (4.16) is satisfied and objective (4.12) is achieved as well. As the number of data packets to be generated in a future interval is unknown at each interval, i.e. \( s_i(k) \forall i \in N, \forall k \) are random values, a routing strategy (to be discussed next) will be proposed to maximize the minimum battery power among all nodes in each interval. This means that in the \( k' \)-th interval, the routing strategy will attempt to achieve the objective (4.19) by choosing the appropriate routes or \( q_i(k') \forall i \in N \) for the given \( r_i(k') \forall i \in N \). Without the knowledge of future packet generation, the original optimization problem discussed in the beginning is transformed into maximizing the minimum battery power or node lifetime among all the nodes in each interval. Since the network lifetime is determined by the node that ends its lifetime first, the effort now is to prolong the network lifetime with respect to the current known packet generation information in each interval.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

\[
\max_{ij} \min(B - (R_i(k^*) + Q_j(k^*))b)
\]  

(4.19)

4.5.2 Max-min Power Credit Routing

A centralized approach, named as max-min Power Credit Routing or \textit{mmPCR}, designed to achieve objective (4.19) is discussed here. Let \(F_{ij}^m\) denotes the \(m\)-th route from source \(i\) to destination \(j\); it is actually a set of forwarding nodes along this route, and \(m \geq 1\). In the static network to be studied, a complete set of routes for all nodes, i.e. \(F_{ij}^m\) \(\forall i, j, m\), is given. Figure 4.11 summarizes the algorithm that will be executed based on the given \(s_i(k) \forall i \in N\) in the \(k\)-th interval. Each source node \(i\) will select only one destination \(j\) in the \(k\)-th interval. A central server is assumed to be present to carry out this computation; it can be the gateway server linking to another network or directly to the internet. Implementing \textit{mmPCR} may not be feasible in mobile and large network where the information to be collected will be overwhelming and this will slow down the routing decision making and dissemination process. The intention here is actually to compare its performance with that of \textit{ODCCR} so as to gain a better understanding of the operation of the former. Thereafter, a distributed and practical version will be investigated as a future work to be done. Consider the \(k\)-th interval. The packets still remaining in the buffer have to be cleared first, starting with the packets carried forward from the first interval \((k^* = 1)\) for all nodes. The clearing of packets will progress from \(k^* = 1\) to \(k\), where the buffered packets carried forward from the \(k^*\)-th interval for all nodes will be attended to progressively. In this manner, the currently generated packets will be served last (at \(k^* = k\)). In each \(k^*\)-th interval, there is a need to find the maximum number of packets \((r_i(k^*, k) = \max_m r_{ij}^m)\) that can be transmitted by each node \(i\).
to its destination \(j\), based on the route cost \((|F_{ij}^m|+1)c\) of each \(m\)-th route and its available credit level \(c_i(k^*,k)\). The number of packets \(r_{ij}^m\) that can be transmitted along the \(m\)-th route from source \(i\) to its destination \(j\) is shown in (4.20). At \(k^* = k\), \(x_i(k^*,k) = s_i(k)\). At the same time, the set of routes \(F_i(k^*,k)\), which comprises routes that allow maximum transmission each, has to be obtained.

\[
\begin{align*}
    r_{ij}^m = \begin{cases} 
    0 & \frac{c_i(k^*,k)}{|F_{ij}^m|+1} \leq 0 \\
    \frac{c_i(k^*,k)}{|F_{ij}^m|+1} & 0 < \frac{c_i(k^*,k)}{|F_{ij}^m|+1} \leq x_i(k^*,k) \\
    x_i(k^*,k) & \frac{c_i(k^*,k)}{|F_{ij}^m|+1} > x_i(k^*,k)
    \end{cases}
\end{align*}
\]

For interval \(k^* = 1\) to \(k\) do

For each source \(i\) and respective destination \(j\), \(\forall m\)

Evaluate \(r_j(k^*,k) = \max_{\forall m} r_{ij}^m\) (based on (4.20))

Find the set of routes \(F_j(k^*,k) = \{F_{ij}^m | r_{ij}^m = r_j(k^*,k)\} \forall m\}

Pre-processing for \(\text{maxminPowerOpt}\) (refer to Figure 4.12)

\(\text{maxminPowerOpt}\) ((4.21) – (4.25)) will output for each \(s \in S(k^*,k)\):

\(f_s^m\), probability of selecting \(m\)'-th route, \(\forall F_{ij}^m \in F_j(k^*,k)\)

Routing and Post-processing after execution of \(\text{maxminPowerOpt}\) (refer to Figure 4.13)

Figure 4.11. Overview of the Operation of max-min Power Credit Routing

Next, pre-processing as shown in Figure 4.12, to prepare the require data for the \(\text{maxminPowerOpt}\) module, will be discussed. For each source node \(i\), if there is a direct route or only one multi-hop route that allows maximum transmission, this is immediately selected and the battery power and credit level of source node \(i\) are decremented appropriately. For the selected multi-hop route, each forwarding node
along this multi-hop route has its battery power decremented and credit level incremented appropriately. In these two cases, this source node will not need the help of the maxminPowerOpt module to obtain its route. If the source node does not have a direct route to its destination, and it has two or more multi-hop routes that allow maximum transmission, $r_i(k^*, k)$ and associated $F_i(k^*, k)$ will be passed to the maxminPowerOpt module for route selection. This source node is added into the set $S(k^*, k)$ and all its possible forwarding nodes into the set $G(k^*, k)$. Lastly, $S(k^*, k), G(k^*, k)$ and the battery power information of all nodes involved will be passed to the maxminPowerOpt module.

For each source $i$

If there is a direct route in $F_i(k^*, k)$
Route is selected

$$b_i(k^*, k) = b_i(k^*, k) - r_i(k^*, k)b$$
$$c_i(k^*, k) = c_i(k^*, k) - r_i(k^*, k)c$$

Else if there is no direct route AND $F_i(k^*, k) = F_{ij}^m$
Route is selected

$$b_i(k^*, k) = b_i(k^*, k) - r_i(k^*, k)b$$
$$c_i(k^*, k) = c_i(k^*, k) - r_i(k^*, k)(|F_{ij}^m| + 1)c$$

For $h \in F_{ij}^m$

$$b_h(k^*, k) = b_h(k^*, k) - r_i(k^*, k)b$$
$$c_h(k^*, k) = c_h(k^*, k) + r_i(k^*, k)c$$

Else if there is no direct route AND $|F_i(k^*, k)| > 1$
Pass $r_i(k^*, k), F_i(k^*, k)$ to maxminPowerOpt
Add $i$ to set $S(k^*, k)$ and all its possible forwarding nodes to $G(k^*, k)$

Pass $b_s(k^*, k) \forall s \in S(k^*, k)$ and $b_g(k^*, k) \forall g \in G(k^*, k)$ to maxminPowerOpt

Figure 4.12. Pre-processing for maxminPowerOpt
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

The aim of maxminPowerOpt is to select routes for all nodes in $S(k^*, k)$ but with the objective of maximizing the minimum battery power among all the forwarding nodes in $G(k^*, k)$ as in (4.19). The formulation is:

$$\max \min_{g \in G} (b_i - (r_i + q_s)b)$$

subject to:

$$q_s = \sum_{g \in F_s, s \in S} r_i f_{s,m} \quad \forall g \in G$$

$$\sum_{s \in S} f_{s,m} = 1 \quad \forall s \in S$$

$$(q_i + r_i)b \leq b_i \quad \forall i \in S \cup G$$

$$f_{s,m} = [0, 1] \quad \forall s \in S, m'$$

($k^*, k$) is removed from all variables to keep the above formulation simple. $f_{s,m}$ is denoted as the probability of source node $s$ selecting the $m'$-th route for transmission, where $F_s m' \in F_s$. For all the participating source nodes, (4.25) shows the possible value that $f_{s,m}$ can assume and (4.23) shows the total probability of using all possible routes (as indicated in $F_s$) adds up to 1. $q_s$ in (4.22) indicates the possible total number of packets a forwarding node $g$ needs to forward by considering all routes that will pass through it. In this optimization exercise, all transmission and forwarding can be done only if the power constraints of all nodes involved are not violated as stated in (4.24). The objective stated in (4.21) will attempt to maximize the minimum battery power of all forwarding nodes; pure source nodes will not be considered in the objective function as the energy expended in self-transmission is a requirement and cannot be changed. The above is solved using MATLAB, with $f_{s,m} \forall s \in S, m'$ as the output computed.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

For post-processing as shown in Figure 4.13, routes are selected for transmission based on the computed output. Considering one packet at a time, a source node \( s \) will choose the \( m^* \)-th route based on the probability \( f_s^{m^*} \). After transmitting a packet, it will then decrement its battery power and credit level. All the forwarding nodes on the selected route will have their battery power and credit levels adjusted as well. Once all source nodes have completed transmission, each source node will update its buffer level. The battery power and credit levels of all nodes will be updated before moving on to clear the buffer at the next \( k^* \)-th interval or to attend to the buffered and newly generated packets at the next \( k \)-th interval. The operation will end once a node has run out of battery power, indicating the end of the operational network lifetime.

For each source \( s \)

For each packet

\( m^* \)-th route is selected for transmission based on the probability \( f_s^{m^*} \)

\[
\begin{align*}
b_{s}(k^*,k) &= b_{s}(k^*,k) - b \\
c_{s}(k^*,k) &= c_{s}(k^*,k) - (\lfloor F_s^{m^*} \rfloor + 1)c
\end{align*}
\]

For \( h \in F_s^{m^*} \)

\[
\begin{align*}
b_{h}(k^*,k) &= b_{h}(k^*,k) - b \\
c_{h}(k^*,k) &= c_{h}(k^*,k) + c \\
x_{i}(k^*,k+1) &= x_{i}(k^*,k) - r_{i}(k^*,k)
\end{align*}
\]

For each node \( i \in N \)

If \( k^* < k \)

\[
\begin{align*}
b_{i}(k^*+1,k) &= b_{i}(k^*,k) \\
c_{i}(k^*+1,k) &= c_{i}(k^*,k)
\end{align*}
\]

Else

\[
\begin{align*}
b_{i}(1,k+1) &= b_{i}(k^*,k) \\
c_{i}(1,k+1) &= c_{i}(k^*,k)
\end{align*}
\]

Figure 4.13. Routing and Post-processing after execution of \textit{maxminPowerOpt}
4.6 Performance Comparison of Two Cost-Credit based Routing Strategies

For performance evaluation, a static network with nodes randomly placed within an area of $3R$ by $3R$, where $R$ is the radio range, based on two node densities: i) 2 nodes per $R$ by $R$ region, and ii) 5 nodes per $R$ by $R$ region, are considered. In each $k$-th interval, each source $i$ will randomly select a destination within three radio ranges, and transmit a random number of packets ($s_i(k)=[1,10]$) within each interval. The initial battery power of each source node $i$ is $b_i(1,1)=B=50$ BUs. Given that $b=c=1$, for $mmPCR$ and $ODCCR$, these routing strategies will be evaluated by varying the initial credit level $c_i(1,1)=C_I$ of each node from $C_I/B = 0.2$ to 1.6. The simulation runs will terminate once the first node has exhausted its battery power, which indicates the end of the network lifetime. The network lifetime (in terms of the number of intervals that have elapsed) and the average number of data packets delivered to the respective destinations within the entire network lifetime against $C_I/B$ are shown in Figure 4.14(a) and 4.14(b) respectively. In these figures, the plots represented by $<protocol>(LD)$ and $<protocol>(HD)$ are obtained from the results of a protocol evaluated in the low node density (2 nodes per $R$ by $R$) and high node density (5 nodes per $R$ by $R$) scenarios respectively. $mmPCR$ is evaluated against the earlier cost-credit routing protocol, $ODCCR$. $AODV$ is included in this study as well so that the two cost-credit routing protocols are also compared with a non cost-credit based routing protocol.
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

Figure 4.14. Performance Comparison of mmPCR, ODCCR and AODV
First, comparing $mmPCR$ and $AODV$, $mmPCR$ is found to outperform $AODV$ when $C/B \geq 0.6$. In terms of the average number of packets carried in the system, $mmPCR$ has a 16% (low density scenario) and 57% (high density scenario) improvement over the amount that $AODV$ can deliver. With a centralized system to manage the power and credit usage of all nodes, the battery power of each node can be used efficiently, thereby improving the network lifetime and the average number of packets carried within the operational network lifetime. With too little initial amount of credits, as in the case of $C/B < 0.6$, the source nodes here take a much longer time to transmit, due to their credit constraints. Therefore, the time till the first node exhausts its battery power is longer. However, the average number of packets carried for $mmPCR$ is about the same as (low density scenario), or greater than (high density scenario) that of $AODV$. $mmPCR$ is able to perform better in a high density scenario because with more forwarding nodes creating more routes, the total forwarding load that determine the battery power usage can be better distributed among more forwarding nodes.

The difference in performance between $ODCCR$ and $mmPCR$ is due to the different objectives set. Both are constrained by the credit system implemented to counter selfish behavior. $ODCCR$ uses a Forwarding Rule (FR) to motivate each node to give up its selfish behavior. It encourages forwarding and still allows a node to transmit as many as self-generated packets as possible. With the FR, each node will forward a number of packets such that it will earn enough credits to transmit using an amount of battery power that is not utilized for forwarding. Once a node has earned enough credits, it will discontinue forwarding and reserve its battery power for future self-transmission. In this manner, each individual node lifetime is extended. From Figure 4.14(a), the network lifetime is found to be significantly longer in the case of $ODCCR$. The average number
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

of packets carried within the network lifetime (Figure 4.14(b)) in the case of ODCCR is higher too due to the longer network lifetime.

When $C/B < 0.6$, the network lifetime increases in the case of ODCCR. This is because with less initial amount of credits given, a node here takes a longer time to earn an amount of credits required to transmit an amount of self-generated packets as specified by its FR. Therefore, each node in this case takes a longer time to exhaust its battery power. Also, with less initial amount of credits given, each node needs to forward more packets, thereby leaving less battery power for its own transmission. Thus, a decrease in average amount of packets carried within the network lifetime as $C/B$ decreases below 0.6 is observed. ODCCR does not perform better than mmPCR when $C/B < 0.6$. This is especially true in the high density scenario where mmPCR has many routes to use for distribution of forwarding load while ODCCR cannot effectively activate FR to control and distribute forwarding load. When $C/B > 1.0$, the network lifetime again increases, but this is due to the decreasing co-operative mode of operation as nodes here have more initial credits to start off with. Nodes here have to wait for a significant amount of time before they can get their multi-hop packets carried through. In the meantime, they can transmit one-hop packets freely as they have enough credits to do so. Figure 4.15 shows the ratio of multi-hop packets transmitted to the total amount of packets transmitted within the network lifetime for the case of ODCCR. The plot here applies to both low node density and high node density scenarios. There is a ‘knee’ at $C/B = 1.0$ and the plot shows a steeply decreasing trend from this point onwards. Operating beyond $C/B = 1.0$ is not recommended as nodes here are sending less amount of multi-hop packets. Even though they are delivering a higher number of packets within the network lifetime, most of these packets are one-hop packets. The recommended
operating condition would be $0.6 \leq C/B \leq 1.0$ for random packet generation. Comparing this with the recommended range for constant packet generation in Section 4.4.1, a typical operation condition recommended would be $0.5 \leq C/B \leq 1.0$.

\textit{mmPCR} is a centralized approach that maximizes the minimum battery power among all nodes in each interval, subjected to the condition that all source nodes must transmit as many packets as possible in each interval. It addresses the lifetime and throughput performance from the network point of view, acting in the interest of the entire network. With the credit system in place to encourage forwarding, \textit{mmPCR} still allows each node to transmit as many packets as possible as allowed by its credit constraint. This implies that if a source node has many routes to its destination but only one of them can allow the most number of packets to be transmitted, this route will be selected without further evaluation, even if this will lead to one or more intermediate nodes depleting all their battery energy. Recall that the number of packets that can be transmitted on a route is dependent on the number of forwarding nodes involved and the available credit level at the source node. When $C/B \geq 0.6$, with enough credits, \textit{mmPCR} allows more routes to be involved in maximizing the minimum battery power found among all the nodes. Thus, \textit{mmPCR} performs much better than \textit{AODV}. However, it does not perform as well as \textit{ODCCR}. This is because while \textit{mmPCR} increases the network throughput and prolongs the network lifetime from the network point of view, it does not address the needs of each individual node. \textit{ODCCR} is better at addressing the needs of each individual node i.e. to forward sufficient amount of packets based on its FR and conserve enough energy so as to increase each individual throughput and lifetime. One interesting feature of \textit{mmPCR} is that, as opposed to other protocols, its performance
becomes better when the node density increases. This is due to the availability of more routes for the centralized system to make a better global decision.

\[\text{Total No. of Multi-hop Packets Transmitted} \div \text{Total No. of Packets Transmitted} \]

Figure 4.15. Ratio of the number of multi-hop packets to the total amount of packets transmitted within the network lifetime in the case of \textit{ODCCR}.

4.7 Conclusion

Nodes are reluctant to forward for others in an ad hoc network because they have limited battery power supply. To handle this problem, an On-Demand Cost-credit routing (\textit{ODCCR}) protocol that enforces forwarding and cooperative routing using a cost-credit concept is proposed. Using a cost-credit model for a simple system such that the nodes in this system can transmit the maximum number of self-generated packets subject to the limits of its battery (power) resource, a Forwarding Rule (FR) is derived. This FR is adapted for implementation in a real system and deployed in the route discovery and management processes of \textit{ODCCR} so as to increase self transmissions of each node in a cooperative and power constrained environment. Performance
Chapter 4 Cost-Credit Based Routing for Cooperation Enforcement

comparisons in static scenarios show that, regardless of the node density, ODCCR gives a higher throughput due to a longer network lifetime, as compared to AODV and RDOBR (Reward based On-demand routing protocol without FR implemented). However, ODCCR will only outperform the other two protocols when an appropriate amount of initial credits is assigned. Since ODCCR uses a route management module that can effectively handle node mobility, simulations show that ODCCR performs as well or better than AODV and RBODR in mobile scenarios.

A centralized max-min Power Credit Routing (mmPCR) protocol is next proposed and its performance is compared with two distributed routing protocols, i.e. AODV and ODCCR. mmPCR performs better than AODV even though the cost-credit system is in place to constrain transmission. This is because mmPCR selects routes for all nodes based on maximizing the minimum battery power among all the nodes, thereby prolonging network lifetime and throughput. At the same time, mmPCR requires all source nodes to transmit as many data packets as possible. However, mmPCR does not perform as well as ODCCR because mmPCR is more focused on addressing performance issues from the network point of view. ODCCR uses the forwarding rule to improve the individual node lifetime and throughput performance. Consequently, the network lifetime and throughput within the network lifetime improves tremendously in the case of ODCCR.
Chapter 5 Location-aided Multi-Objective Routing

5.1 Introduction

In a truly ad hoc environment, it may not be possible to replenish the limited battery supply of the participating nodes. As mentioned in Chapter 3, this implies that routing based on power awareness factors is more important than routing based on smallest hop counts in a power constrained environment. For this, there are two power awareness factors to be considered when selecting a route - maximizing the minimum node battery power and minimizing the total transmission power required to reach a destination.

Therefore, Section 5.2 first proposes a general power aware multi-objective problem and follows this with three variations which would be of interest. These power aware multi-objective problems can be solved using standard optimization techniques. However, carrying out this optimization is not feasible in a large and mobile network where solutions to the routing problems need to be obtained promptly. This motivates the design of three corresponding heuristics which can be practically deployed in the proposed on-demand Location-aided Multi-objective Routing protocol (Power Aware) or LAMOR-PA, which will be described in Section 5.3. Note that here each node is assumed to know its current location via GPS or some other appropriate localization techniques. Using this location information, a flood control mechanism is also proposed and discussed in the same section. The results of the performance studies conducted on LAMOR-PA in static networks are also presented in Section 5.3.

In a truly ad hoc environment, nodes as individuals will move out of the network as and when they need to. This will affect ongoing communications taking place or routed via
them (as forwarders). Therefore, route lifetime will be affected by node mobility and needs to be factored in when selecting routes. Three variations of a power and mobility aware multi-objective problem with associated constraints are formulated and presented in Section 5.4. This motivates the design of three corresponding on-demand Location-aided Multi-objective Routing protocols (Power and Mobility Aware) or LAMOR-PMA, which are discussed in Section 5.5. Since each node only knows its location, and is unlikely to have additional equipment to obtain its speed and heading direction, a motion prediction module is proposed to obtain the latter two parameters. With all three parameters known, the link and route lifetime can then be computed. In the same section, the performance results of LAMOR-PMA when evaluated in mobile networks are presented. Section 5.6 concludes this chapter.

5.2 Power Aware Multi-Objective Routing

5.2.1 General Problem Formulation

Let $s$ be the source, and $F^m$ be the set consisting of the source $s$ and the forwarding nodes located along the $m^{th}$ route to a destination. $e^m_g$ is denoted as the amount of transmission energy required to send a bit of data from a node $g$ to the next downstream node that is at a distance $d^m_g$ away, along the $m^{th}$ route, $g \in F^m$. The transmission energy model, which is reflected in (5.1), is the same as that used in [30] and [85]. In (5.1), $E_{elec}$ is required to run the transceiver circuitry and has a value of 50nJ/bit. $E_{amp}$ is required for the transmit amplifier to achieve an acceptable signal-to-noise ratio at the receiver that is 1 metre away; it has a value of 100pJ/bit/m$^2$. If the receiver is located at a distance $d_m$ away, the transmit amplifier needs to expend $E_{amp}d^2_m$ for a bit to be
Chapter 5 Location-aided Multi-Objective Routing

received correctly at the receiver. The energy consumed for reception is very small as compared to the energy required for transmission and is neglected [31], [77], [82].

\[ e^m_g = E_{amp} (d^m_g)^2 + E_{elec} \quad (5.1) \]

Let \( b_g \) be the residual energy of each node \( g \). A general power aware multi-objective formulation is shown below with (5.2) and (5.3) being the two power aware objectives and (5.4) to (5.8) being the associated constraints. \( b^m \) in (5.4) gives the minimum available battery energy among all the nodes along the \( m \)th route. \( e^m \) in (5.5) denotes the total transmission energy required for transmitting one bit of data along the \( m \)th route. The objective here is to select a route based on minimizing the total transmission energy required as in (5.2) and maximizing the minimum remaining battery energy as in (5.3). Constraint (5.6) states that if at least one node has exhausted its battery power resource, a path that goes through any of these nodes cannot be selected. Given that \( f^m \) corresponds to the \( m \)th route, it assumes the value of 0 (to indicate it is not selected) or 1 (selected) as in (5.8). Constraint (5.7) states that only one route will be chosen.

\[
\begin{align*}
\min & \sum_{m} e^m \ f^m & \quad (5.2) \\
\max & \sum_{m} b^m \ f^m & \quad (5.3) \\
\text{subject to:} & \\
& b^m = \min_{g \in F^m} b_g \quad \forall m & \quad (5.4) \\
& e^m = \sum_{g \in F^m} e^m_g \quad \forall m & \quad (5.5) \\
& b^m f^m > 0 \quad \forall m & \quad (5.6) \\
& \sum_{m} f^m = 1 & \quad (5.7)
\end{align*}
\]
Chapter 5 Location-aided Multi-Objective Routing

\[ f^m = [0,1] \ \forall m \quad (5.8) \]

Some methods for solving multi-objective optimization problem are proposed in [86]. The commonly used \textit{weighting method} transforms (5.2) and (5.3) into (5.9) and then solves it as a binary integer programming problem, subject to (5.4) - (5.8).

\[
\min(w_1(\sum_{\forall m} e^m f^m) - w_2(\sum_{\forall m} b^m f^m)) \quad (5.9)
\]

Normalizing each objective function (5.2) and (5.3) by their respective maximum values \((e_{\text{max}}, b_{\text{max}})\) in (5.9) yields (5.10). This normalization transforms (5.2) and (5.3) so that each will assume comparable values in \([0,1]\) to prevent any one of them from dominating the minimization process. In (5.9), it is mentioned in [86]-[87] that the actual value of each weight \((w_1, w_2)\) should depend on the relative importance of the corresponding objective function (5.2) or (5.3). As each objective function ((5.2) or (5.3)) is equally important, and a routing decision based on balancing the two requirements as stated in the two objective functions needs to be obtained, \(w_1 = w_2 = 1\) will be set in (5.10).

\[
\min((\sum_{\forall m} \frac{e^m f^m}{e_{\text{max}}}) - (\sum_{\forall m} \frac{b^m f^m}{b_{\text{max}}})) \quad (5.10)
\]

5.2.2 Variations with Load-Dependent Formulations

Various important variations to the general power aware multi-objective problem of Section 5.2.1 have been explored. Two variations that are \textit{Load-Dependent Formulations} are presented here. The first variation considers the data traffic load...
Chapter 5 Location-aided Multi-Objective Routing

experienced or energy consumption in the past interval based on the premise that it
would not be a good idea to request a node to forward packets if it has been loaded
either by self-generated or forwarded data traffic recently. Instead, this node should be
recruited to forward in later intervals when it is not heavily utilized. This will ensure
that nodes take turns to forward packets for others. Let $c_g$ be the energy consumption of
each node $g$ over a past fixed interval which is appropriately determined. Among all the
nodes along the $m^{th}$ route, $c^m$ in (5.11) then gives the maximum energy consumption of
a node over the past interval. In this variation, route selection based on minimization of
the maximum past energy consumption as in (5.12) needs to be considered as well. To
incorporate this, (5.12) is normalized by its maximum value $c_{\text{max}}$ and then added as an
additional objective to (5.10) yielding (5.13) as the resulting objective function. For this
load dependent formulation, the objective function of (5.13) is used with associated
constraints (5.4) - (5.8) and (5.11).

$$c^m = \max_{\forall g, F^m} c_g \quad \forall m$$  \hspace{1cm} (5.11)

$$\min \sum_{\forall m} c^m f^m$$  \hspace{1cm} (5.12)

$$\min\left(\frac{\sum c^m f^m}{e_{\text{max}}} - \frac{\sum b^m f^m}{b_{\text{max}}} + \frac{\sum c^m f^m}{c_{\text{max}}})\right)$$  \hspace{1cm} (5.13)

The second variation considers the forwarding load experienced or the energy utilized
so far for forwarding. The rationale here is that if one node has not forwarded much for
others, it should make itself more available for forwarding, even if it needs to transmit
large amounts of self-generated data. This will ensure that each node has a good chance
to deliver a fair amount of its data across the network. Let $u_g$ be the ratio of the amount
of energy used for forwarding to the total energy used at node $g$ so far. $u^m$ in (5.14)
Chapter 5 Location-aided Multi-Objective Routing

gives, among all the nodes along the $m^{th}$ route, the maximum ratio of the amount of energy used for forwarding to the total energy used at the node. In the second variation, route selection based on minimizing the maximum ratio of the amount of energy used for forwarding to the total energy used as in (5.15) needs to be considered as well. Here, (5.15) will be normalized by its maximum value $u_{\text{max}}$ and then added as an additional objective to (5.10) to yield the resulting objective function of (5.16). For this forwarding load dependent formulation, the objective function of (5.16) is used with associated constraints (5.4) - (5.8) and (5.14).

$$u^m = \max_{\forall s \in F^m} u^m \quad \forall m$$  \hspace{1cm} (5.14)

$$\min \sum_{\forall m} u^m f^m$$ \hspace{1cm} (5.15)

$$\min((\frac{\sum_{\forall m} e^m f^m}{e_{\text{max}}}) - (\frac{\sum_{\forall m} b^m f^m}{b_{\text{max}}}) + (\frac{\sum_{\forall m} u^m f^m}{u_{\text{max}}}))$$ \hspace{1cm} (5.16)

5.2.3 Variation with Known Transmission Requirement

The third variation of the basic formulation considers a situation where the transmission requirements are given in the form of the amount of data to be transmitted during a connection. This would be applicable to applications like file and video downloads where the size of data to be transmitted is known a priori. With the transmission requirement given, the current load a node is experiencing, or its current energy consumption from ongoing connections that it has originated and those that are being forwarded through it, can be obtained. This will allow the current energy consumption of a node to be added as an additional consideration when selecting routes. (Note that the earlier variation of Section 5.2.2 only considers past energy consumption.)
Chapter 5 Location-aided Multi-Objective Routing

The formulation here comprises the objective function (5.17) and its associated constraints (5.18) to (5.23). Constraint (5.18) is similar to (5.5). $e_g$ in (5.19) is denoted as the total transmission energy that is required of a node $g$ to transmit or forward the remaining number $q^l_g$ of data bits for each of the ongoing connection along the $l^{th}$ route, $l \in L_g$. $L_g$ is the set of routes currently routed via node $g$ and the route currently used by this node to transmit its generated data bits. $e^l_g$ is the transmission energy that node $g$ is required to expend to transmit one data bit along the $l^{th}$ route. The source node and the forwarding nodes should have enough energy resources to support the entire transmission for the route selected. Consider the $m^{th}$ route. Here $b^m$ in (5.20) gives, among all the nodes along this route, the minimum difference between the battery energy available (after deducting an amount required for ongoing transmission) and the transmission energy required to transmit $r$ amount of data. $b^m$ in (5.20) is considered the minimum effective remaining battery energy found among all forwarding nodes and the source node along the $m^{th}$ route. For the $m^{th}$ route to be considered for selection, $b^m$ must be greater than or equal to 0, as enforced by constraint (5.21). Constraints (5.22) and (5.23) enforce that only one minimum cost route is selected. The objective (5.17) here is to select a route based on minimizing the total transmission energy and maximizing the minimum effective remaining battery energy found among all forwarding nodes.

$$\min \left( \sum_{\forall m} b^m f^m \right) - \left( \sum_{\forall m} b^m f^m \right)$$

subject to:

$$e^m = \sum_{g \in F^n} e^m_g \quad \forall m$$

$$e_g = \sum_{\forall l \in L_g} q^l_g e^l_g \quad \forall g$$

(5.17)

(5.18)

(5.19)
Chapter 5 Location-aided Multi-Objective Routing

\[ b^m = \min_{\forall g \in F^m} (b_s - e_g - re_g^m) \quad \forall m \quad (5.20) \]

\[ b^m f^m \geq 0 \quad \forall m \quad (5.21) \]

\[ \sum_{\forall m} f^m = 1 \quad (5.22) \]

\[ f^m = [0,1] \quad \forall m \quad (5.23) \]

Let \( r_s \) be the total data size to be transmitted by a source in a connection. The above formulation can be solved as a binary integer programming problem, with the intention to transmit an amount \( r = r_s \) from source. If this yields no solution, \( r = \min_{\forall m} \left( \max_{\forall g \in F^m} \left( \frac{b_s - e_g}{e_g^m}, r_s - r_{\text{accum}} \right) \right) \) is set and the problem is solved again to obtain a route to transmit the amount \( r \). After transmitting, \( r_{\text{accum}} \), which is initialized to zero and will accumulate the amount of data sent so far, will be updated. The optimization process will then be repeated if there are still data to be transmitted, i.e., if \( r_{\text{accum}} < r_s \).

5.3 On-demand Location-aided Power Aware Routing

5.3.1 Flood Control Mechanism

In order to incorporate the multi-objective routing objectives discussed in the previous section in actual wireless ad hoc networks, three corresponding on-demand location aided routing protocols are proposed. (As stated earlier, each node is assumed to be able to obtain its current location either via GPS or some other localization technique.) Being on-demand routing protocols, a source node will perform route discovery by flooding the network with route request packets or RREQs. This will then allow the source node to obtain a route for transmission upon receiving the route information carried by the
route reply packets or RREPs. To control flooding, a flood control mechanism using the available location information is proposed. The flood control mechanism comprises two RREQ filtering steps. The basic concepts of the flood control mechanism are presented here; the actual operation is subsequently explained in Section 5.3.2.

The first RREQ filtering step restricts RREQ flooding within a broadcast area or forwarding zone. LAR [33] or PMLAR [34] employs a similar restricted broadcast area to restrict RREQ flooding. While both LAR and PMLAR consider the speed of the destination, both the speed and pause time of the destination are considered in order to derive the forwarding zone here. In addition, the proposed strategy here for determining the forwarding zone allows enough routes to be discovered so that a better overall routing decision can be made. When a source node needs to determine the forwarding zone, it will retrieve the last known information regarding the destination $d$ and compute $r_{\text{zone} \_d}$, the radius of the destination zone as in (5.24).

$$r_{\text{zone} \_d} = v_d (t'_d - t^0_d) \quad (5.24)$$

where $r_{\text{zone} \_d}$ is the radius of the destination zone of destination $d$

$v_d$ is the last known speed of destination $d$

$t^0_d$ is the time when $v_d$ and location $(x_d, y_d)$ of $d$ are collected

$t'_d$ is the time that the computation of $r_{\text{zone} \_d}$ needs to be done

Equation (5.24) can be used only if destination $d$ is still in motion when its speed $v_d$ and location $(x_d, y_d)$ need to be collected. If $d$ is stationary when its motion profile is required, (5.25) will be used instead. Such information is typically collected when the destination is constructing the RREP for transmission towards the source.
Chapter 5 Location-aided Multi-Objective Routing

\[
r_{\text{zone}d} = \begin{cases} 
  v_d(t_d^1 - t_d^0 - \overline{tp_d^0}) & \text{if } (t_d^1 - t_d^0) \geq \overline{tp_d^0} \\
  0 & \text{if } (t_d^1 - t_d^0) < \overline{tp_d^0}
\end{cases}
\]  

(5.25)

where \( \overline{v_d} \) is the weighted average speed of destination \( d \) as obtained using (5.26)

\( \overline{tp_d^0} \) is effective weighted average pause time. It is the weighted average pause time \( \overline{tp_d} \) of destination \( d \) as obtained using (5.27) minus the amount of time \( t_{\text{motionless}} \) that has elapsed since \( d \) has been motionless if \( \overline{tp_d} \geq t_{\text{motionless}} \). It assumes a zero value if \( \overline{tp_d} < t_{\text{motionless}} \).

For each node \( i \), it will record its weighted average speed and pause time as represented by (5.26) and (5.27) respectively.

\[
\overline{v_i} = \alpha \overline{v_i}^{\prime} + (1 - \alpha) v_i
\]

(5.26)

\[
\overline{tp_i} = \alpha \overline{tp_i}^{\prime} + (1 - \alpha) tp_i
\]

(5.27)

\( \overline{v_i}^{\prime} \) and \( \overline{tp_i}^{\prime} \) are the weighted average speed and pause time of node \( i \) before it is updated with the recent \( v_i \) and \( tp_i \) values respectively. \( \overline{v_i} \) and \( \overline{tp_i} \) are the weighted average speed and pause time of node \( i \) after the node is updated with the recent \( v_i \) and \( tp_i \) values respectively. \( \alpha \) assumes a value between 0 and 1. \( \alpha=0.99 \) is set here so that the weighted average values will not be affected too much by sudden transient changes in recently acquired values. The study of the actual impact of setting different values of \( \alpha \) will be a future work to be done.

The size of the forwarding zone is at least \( R \) by \( R \), where \( R \) is the radio range. This allows enough number of nodes to be found, especially if multiple route options are required for the selection of the least cost route. (Note that this is a major departure from
Chapter 5 Location-aided Multi-Objective Routing

LAR and PMLAR, which do not use such a minimum size. In LAR and PMLAR, as long as one route can be found, it will be selected. They resort to flooding the entire area if a route cannot be found within their restricted broadcast areas.) The proposed forwarding zone here will be obtained as shown in the approach summarized in Figure 5.1. Figure 5.2 shows an example of the proposed forwarding zone. In Figure 5.2, if the intermediate node finds itself to be within the forwarding zone, it will forward the RREQ; otherwise it will drop the RREQ. For example, in Figure 5.2, node $i$ will forward the RREQ while node $j$ will not.

The second RREQ filtering step requires an intermediate node to forward the RREQ only if it finds that it will make positive progress towards, or is nearer to, the destination. Each node here knows its current location. Location information of the destination and the immediate upstream neighbor will be included in the RREQ. This way each intermediate node is able to find out whether it is nearer to the destination as compared to the upstream neighbor. Using Figure 5.3 as an illustration, node $i$ will broadcast the RREQ from node $j$ as it is nearer to the destination $d$ than node $j$. However, node $k$ will drop the RREQ from node $j$ as it is further away from the destination than node $j$. 
Given the coordinates of source $s(x_s, y_s)$ and destination $d(x_d, y_d)$

The coordinates of the 4 edges of the forwarding zone are denoted as:

- Bottom Left Edge: $\{x_{BLE}, y_{BLE}\}$
- Top Left Edge: $\{x_{TLE}, y_{TLE}\}$
- Bottom Right Edge: $\{x_{BRE}, y_{BRE}\}$
- Top Right Edge: $\{x_{TRE}, y_{TRE}\}$

Let

\[ x_{min} = \min(x_s, x_d - r_{\_zone_d}) \]
\[ x_{max} = \max(x_s, x_d + r_{\_zone_d}) \]
\[ y_{min} = \min(y_s, y_d - r_{\_zone_d}) \]
\[ y_{max} = \max(y_s, y_d + r_{\_zone_d}) \]

The edges of the forwarding zone are computed as follows:

If $x_{max} - x_{min} < R$
\[ \text{diff}_x = (R - x_{max} + x_{min})/2 \]
\[ x_{BLE} = x_{TLE} = x_{min} - \text{diff}_x \]
\[ x_{BRE} = x_{TRE} = x_{max} + \text{diff}_x \]
Else
\[ x_{BLE} = x_{TLE} = x_{min} \]
\[ x_{BRE} = x_{TRE} = x_{max} \]

If $y_{max} - y_{min} < R$
\[ \text{diff}_y = (R - y_{max} + y_{min})/2 \]
\[ y_{BLE} = y_{BRE} = y_{min} - \text{diff}_y \]
\[ y_{TLE} = y_{TRE} = y_{max} + \text{diff}_y \]
Else
\[ y_{BLE} = y_{BRE} = y_{min} \]
\[ y_{TLE} = y_{TRE} = y_{max} \]

Figure 5.1. Computation to obtain the Forwarding Zone
5.3.2 Route Request Phase

Three on-demand location-aided multi-objective routing protocols LAMOR-PAL, LAMOR-PAF and LAMOR-PA/TR are proposed as heuristic practical implementations.
Chapter 5 Location-aided Multi-Objective Routing

for the multi-objective power aware approaches of Section 5.2, i.e. (a) load dependent power aware, (b) forwarding load dependent power aware and (c) power aware with known transmission requirements, respectively. The route discovery (route request and route reply phases) process to be discussed will be common to all three protocols. When a source node needs to discover a route for transmission, it will construct a RREQ with the following common structure: 

\[(src\_addr, dest\_addr, (x_d, y_d), sequence\_no, F\_ZONE, e\_tot, (x_{nb}, y_{nb}), F\_SEQ)\]

The definitions of these fields are given in Table 5.1. Each of the three \textit{LAMOR-PA} protocols will also have different additional fields as indicated in Table 5.1.

Figures 5.4, 5.5 and 5.6 summarize the Route Request Processes at intermediate nodes for \textit{LAMOR-PAL}, \textit{LAMOR-PAF} and \textit{LAMOR-PA/TR} respectively. When an intermediate node intercepts the RREQ, it will perform the checks given below. These checks will reduce RREQ traffic and will also aid in selecting the routes at the destination node. The intermediate node will check against the previous record in the RREQ_Table and update the entry if necessary. The structure of this table is shown in Table 5.2. Fields pertaining to a specific protocol have been specifically indicated in the table.

a. Is \(sequence\_no\) outdated, i.e. \(sequence\_no(RREQ) < sequence\_no(RREQ\_Table)\)?

b. Is the current node the destination node?

c. Is the current node located within the Forwarding Zone?

d. Is the current node further away from the destination node, as compared to the immediate upstream neighbor that sent the RREQ to it?

e. Are the energy constraints satisfied (applies only to \textit{LAMOR-PA/TR})?
Chapter 5 Location-aided Multi-Objective Routing

f. Is the route cost carried by RREQ lower than the route cost recorded in the RREQ_Table?

Table 5.1. Definition of the various fields in RREQ

<table>
<thead>
<tr>
<th>Fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>src_addr</td>
<td>Address of the source node $s$ initiating a route request.</td>
</tr>
<tr>
<td>dest_addr</td>
<td>Address of the destination node $d$.</td>
</tr>
<tr>
<td>$(x_d, y_d)$</td>
<td>Last known co-ordinates of node $d$ as recorded in source node.</td>
</tr>
<tr>
<td>sequence_no</td>
<td>Unique sequence number pertaining to a route request. This is set by the source node only to identify each unique route request.</td>
</tr>
<tr>
<td>F_ZONE</td>
<td>Forwarding zone. The source node will define this area as discussed in Section 5.3.1.</td>
</tr>
<tr>
<td>$e_{tot}$</td>
<td>Records the total transmission power as the RREQ propagates from the source to the destination node. This is initialized to zero by the source.</td>
</tr>
<tr>
<td>$(x_{nb}, y_{nb})$</td>
<td>Records the location of immediate upstream neighbor. The source node will initialize with its own location.</td>
</tr>
<tr>
<td>F_SEQ</td>
<td>Records the sequence of forwarding nodes between the source node and the destination node. It is an empty set initially.</td>
</tr>
<tr>
<td>load_max</td>
<td>Records the maximum node energy consumption in the past interval as the RREQ propagates from the source to the destination node. This is initialized to zero by the source. Available in LAMOR-PAL only.</td>
</tr>
<tr>
<td>forward_load_max</td>
<td>Records the maximum ratio of the amount of energy used for forwarding to the total energy used thus far in a node as the RREQ propagates from the source to the destination node. This is initialized to zero by the source. Available in LAMOR-PAF only.</td>
</tr>
<tr>
<td>$b_{min}$</td>
<td>Records the minimum node battery power as the RREQ propagates from the source to the destination node. This is initialized with the source current battery level by the source. Applies to LAMOR-PAL and LAMOR-PAF only.</td>
</tr>
<tr>
<td>$r$</td>
<td>Records the number of data packets the source will transmit in this connection to be made. Available in LAMOR-PA/TR only.</td>
</tr>
<tr>
<td>$r_{max}$</td>
<td>The maximum number of data packets that can be transmitted along the route built. The source node will initialize with a value of zero. Available in LAMOR-PA/TR only.</td>
</tr>
<tr>
<td>$b_{nb}$</td>
<td>Records the remaining battery power of the immediate upstream neighbor. The remaining battery power of the node here is its current remaining battery power level less the amount reserved for the ongoing transmissions. The source node will initialize with its remaining battery power. Available in LAMOR-PA/TR only.</td>
</tr>
<tr>
<td>$b_{eff_min}$</td>
<td>Records the minimum effective node battery power as the RREQ propagates from the source to the destination node. The effective node battery power here is the node’s remaining battery power level minus the transmission power required to deliver $r$ amount of data to the next hop neighbor. This is initialized to zero by the source. Applies to LAMOR-PA/TR only.</td>
</tr>
</tbody>
</table>
Table 5.2. Definition of the various fields in RREQ_Table

<table>
<thead>
<tr>
<th>Fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>src_addr</td>
<td>Address of the source node $s$ initiating a route request.</td>
</tr>
<tr>
<td>sequence_no</td>
<td>A unique number identifying a particular route request from the source node.</td>
</tr>
<tr>
<td>$e_{tot}$</td>
<td>The total transmission power required for the route between the source node and the node that records the route request information.</td>
</tr>
<tr>
<td>$max_e_{tot}$</td>
<td>The maximum total transmission power required, found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$.</td>
</tr>
<tr>
<td>$F_{SEQ}$</td>
<td>The sequence of forwarding nodes between the source node and the node that records the route request information.</td>
</tr>
<tr>
<td>load_max</td>
<td>The maximum node energy consumption in the past interval found on a route between the source node and the node that records the route request information. Available in LAMOR-PAL only.</td>
</tr>
<tr>
<td>$max_load_{max}$</td>
<td>The maximum value of the maximum node energy consumption in the past interval found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$. Available in LAMOR-PAL only.</td>
</tr>
<tr>
<td>forward_load_max</td>
<td>The maximum ratio of the amount of energy used for forwarding to the total energy used thus far in a node found on a route between the source node and the node that records the route request information. Available in LAMOR-PAF only.</td>
</tr>
<tr>
<td>$max_forward_load_{max}$</td>
<td>The maximum value of the maximum ratio of the amount of energy used for forwarding to the total energy used thus far in a node found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$. Available in LAMOR-PAF only.</td>
</tr>
<tr>
<td>$b_{min}$</td>
<td>The minimum node battery power found on a route between the source node and the node that records the route request information. Available in LAMOR-PAL and LAMOR-PAF only.</td>
</tr>
<tr>
<td>$max_b_{min}$</td>
<td>The maximum value of the minimum node battery power found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$. Available in LAMOR-PAL and LAMOR-PAF only.</td>
</tr>
<tr>
<td>$r_{max}$</td>
<td>The maximum number of data packets that can be transmitted along the route built and seen so far, pertaining to node $s$. Applies to LAMOR-PA/TR only.</td>
</tr>
<tr>
<td>$b_{eff}_{min}$</td>
<td>The minimum effective node battery power found on a route between the source node and the node that records the route request information. Available in LAMOR-PA/TR only.</td>
</tr>
<tr>
<td>$max_b_{eff}_{min}$</td>
<td>The maximum value of the minimum effective node battery power found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$. Available in LAMOR-PA/TR only.</td>
</tr>
</tbody>
</table>
intermediate $k$:

if (src_addr, sequence_no) of RREQ is the same OR more updated
then
  CHK_ROUTE_FN = FALSE

if $k$ is the destination $d$
then CHK_ROUTE_FN = TRUE
else if $k$ is NOT the source $s$ OR destination $d$
then if $k$ is in $F\_ZONE$(RREQ)
  then if $k$ is nearer to $d$ than neighbor $nb$
    then CHK_ROUTE_FN = TRUE

if (CHK_ROUTE_FN = TRUE)
then
  update $e\_tot$(RREQ)
  update in RREQ_Table: max$_e\_tot$, max$_b\_min$ and max$_load\_max$

if $c(m,s,k)$(RREQ) $\geq$ $c(m',s,k)$(RREQ_Table)
then exit processing and drop RREQ

update in RREQ_Table from RREQ: $e\_tot$, $b\_min$, $load\_max$, $F\_SEQ$

if $k$ is NOT the destination $d$
then
  update in RREQ from node: $b\_min$, $load\_max$, $(x_{nb}, y_{nb})$, $F\_SEQ$
  rebroadcast RREQ
else if $k$ is the destination $d$
then if (src_addr, sequence_no) of RREQ is more updated
  then set timer to collect routes

Figure 5.4. Route Request Process for LAMOR-PAL
intermediate $k$:

if $(\text{src\_addr, sequence\_no})$ of RREQ is the same OR more updated
then
CHK\_ROUTE\_FN = FALSE

if $k$ is the destination $d$
then CHK\_ROUTE\_FN = TRUE
else if $k$ is NOT the source $s$ OR destination $d$
then if $k$ is in $F\_ZONE$(RREQ)
then if $k$ is nearer to $d$ than neighbor $nb$
then CHK\_ROUTE\_FN = TRUE

if (CHK\_ROUTE\_FN = TRUE)
then
update $e\_tot$(RREQ)
update in RREQ\_Table:
$\text{max\_e\_tot, max\_b\_min and max\_forward\_load\_max}$

if $c(m,s,k)\ (RREQ) \geq c(m',s,k)\ (RREQ\_Table)$
then exit processing and drop RREQ

update in RREQ\_Table from RREQ:
$e\_tot, b\_min, \text{forward\_load\_max, F\_SEQ}$

if $k$ is NOT the destination $d$
then
update in RREQ from node:
$b\_min, \text{forward\_load\_max}, (x_{nb}, y_{nb}), F\_SEQ$
rebroadcast RREQ
else if $k$ is the destination $d$
then if $(\text{src\_addr, sequence\_no})$ of RREQ is more updated
then set timer to collect routes

Figure 5.5. Route Request Process for $\text{LAMOR-PAF}$
intermediate \( k \):

\[
\text{if } (\text{src_addr, sequence_no}) \text{ of RREQ is the same OR more updated }
\]
\[
\text{then}
\]
\[
\text{CHK_ROUTE_FN = FALSE}
\]
\[
\text{if } k \text{ is the destination } d
\]
\[
\text{then } \text{CHK_ROUTE_FN = TRUE}
\]
\[
\text{else if } k \text{ is NOT the source } s \text{ OR destination } d
\]
\[
\text{then if } k \text{ is in } F_ZONE(\text{RREQ})
\]
\[
\text{then if } k \text{ is nearer to } d \text{ than neighbor } nb
\]
\[
\text{then } \text{CHK_ROUTE_FN = TRUE}
\]

\[
\text{if } (\text{CHK_ROUTE_FN = TRUE})
\]
\[
\text{then}
\]
\[
\text{update } b_{\text{eff_min}}(\text{RREQ}), e_{\text{tot}}(\text{RREQ})
\]
\[
\text{if } k \text{ is NOT the destination } d
\]
\[
\text{then}
\]
\[
\text{if } (\text{check_constraints_module (See Figure 5.7) = TRUE})
\]
\[
\text{then if } c(m, s, k)(\text{RREQ}) \geq c(m', s, k)(\text{RREQ}_\text{Table})
\]
\[
\text{then exit processing and drop RREQ}
\]
\[
\text{update in } \text{RREQ}_\text{Table} \text{ from } \text{RREQ}: e_{\text{tot}}, b_{\text{eff_min}}, F_{\text{SEQ}}
\]
\[
\text{update in } \text{RREQ} \text{ from node}: b_{nb}, (x_{nb}, y_{nb}), F_{\text{SEQ}}
\]
\[
\text{rebroadcast } \text{RREQ}
\]

\[
\text{else if } k \text{ is the destination } d
\]
\[
\text{then}
\]
\[
\text{build routes (See Figure 5.8)}
\]
\[
\text{if } (\text{src_addr, sequence_no}) \text{ of RREQ is more updated }
\]
\[
\text{then set timer to collect routes}
\]

Figure 5.6. Route Request Process for LAMOR-PA/TR

The Route Request Processes at intermediate nodes for LAMOR-PAL and LAMOR-PAF are discussed first; this process is the same for both protocols except for the route costs used. The first step (a) taken by the intermediate node is to check if the RREQ is outdated. If the RREQ is not outdated, it will check if it is the destination node itself. An outdated RREQ will be dropped. In step (b), if the node is itself the destination node, then steps (c) and (d) will be skipped and it will proceed directly to step (f). (Step (e)
does not apply to LAMOR-PAL and LAMOR-PAF.) Otherwise, the intermediate node will go through steps (c) and (d) to find out if it is required to process the RREQ for possible broadcasting later. These two steps have been discussed in Section 5.3.1 as part of the proposed flood control mechanism. In step (f), the route carried in the RREQ will be compared against the route residing in the RREQ_Table based on their respective route costs. The route costs \( c(m,s,i) \) for the \( m \)th route from source \( s \) to current node \( i \) for LAMOR-PAL and LAMOR-PAF are given in (5.28) and (5.29) respectively. These are based on their respective objective functions (5.13) and (5.16) respectively. The definitions of the different terms in (5.28) and (5.29) can be found in Table 5.1 and 5.2. For the route carried by the RREQ, the transmission power \( \text{transmit\_power} \) between the immediate upstream neighbor and this intermediate node can be determined from the power received, or from the positions of these two nodes using (5.1). Thereafter, this transmission power value is added to \( e\_tot(RREQ) \) to obtain the total transmission power required should this route be used.

**LAMOR-PAL:**

\[
c(m,s,i) = \left(\frac{e\_tot}{\max\_e\_tot}\right) - \left(\frac{b\_min}{\max\_b\_min}\right) + \left(\frac{load\_max}{\max\_load\_max}\right)
\]  
(5.28)

**LAMOR-PAF:**

\[
c(m,s,i) = \left(\frac{e\_tot}{\max\_e\_tot}\right) - \left(\frac{b\_min}{\max\_b\_min}\right) + \left(\frac{forward\_load\_max}{\max\_forward\_load\_max}\right)
\]  
(5.29)

The \( \max\_e\_tot, \max\_b\_min, \max\_load\_max \) and \( \max\_forward\_load\_max \) fields in the RREQ_Table of the node record maximum total transmission power, maximum value of the minimum node battery power, maximum value of the maximum node energy consumption in the past interval and maximum value of the maximum ratio of the
amount of energy used for forwarding to the total energy used thus far in a node respectively. These fields are updated each time a RREQ is encountered. If the route cost function of the route carried by the RREQ does not have a lower value, the RREQ will be dropped. Otherwise, the node will continue to the last stage of RREQ processing—updating and re-broadcasting.

In the final stage, the values of the $e_{tot}$, $b_{min}$, $load_{max}$(LAMOR-PAL) and $forward_{load}_{max}$(LAMOR-PAF) fields in the RREQ_Table will be replaced by the corresponding values obtained from the RREQ. In addition, the sequence of forwarding nodes from $F_{SEQ}$(RREQ), which is the set of forwarding nodes between the source node and the current intermediate node, will be copied to the RREQ_Table. If the route cost function of the route carried by the RREQ has a lower value, the node needs to rebroadcast the RREQ if it is not the destination. Before doing so, it will update $b_{min}$(RREQ) if it finds that it has a lower battery power level. $load_{max}$(RREQ) and $forward_{load}_{max}$(RREQ) will be updated in the case of LAMOR-PAL and LAMOR-PAF respectively, if the node finds that it has higher corresponding values. The node will also update its location in the field ($x_{nb}$, $y_{nb}$) of the RREQ. Lastly, it will add its IP address into $F_{SEQ}$(RREQ). If the given node is the destination node, and the RREQ received is the first RREQ it receives from a source in a route discovery cycle, the node will wait for some period of time to collect and evaluate the routes carried in the RREQ before it decides on the final route to use.

For LAMOR-PA/TR the Route Request Process at intermediate nodes is quite similar to that described for the former two LAMOR-PA protocols. The main difference here is that nodes will need to go through step (e) that checks battery energy or power
constraints (5.21) with the transmission requirements provided. The details of the constraint checking process are summarized in Figure 5.7. For the route carried by the RREQ, the transmission power $\text{transmit\_power}$ between the immediate upstream neighbor and this intermediate node can be obtained from the power received, or from the positions of these two nodes using (5.1). Thereafter, the value of $\text{transmit\_power}$ is added to $e_{tot}(\text{RREQ})$ to obtain the total transmission power required, should this route be used. With the remaining battery power $b_{nb}(\text{RREQ})$ of this neighbor provided, the maximum amount of data packets $r_{max\_b}$ the neighbor can send can be determined. Subsequently, the effective maximum amount of data packets $r_{eff}$ that the route in the RREQ can carry can be determined too. The remaining battery power of the node in the case of LAMOR-PA/TR is the current remaining battery power level less the amount reserved for the ongoing transmissions. For LAMOR-PA/TR, each node is able to reserve an amount for ongoing transmissions since the transmission requirement or the number of data packets to be transmitted in a connection is given a priori by the source. $b_{eff\_min}$ field of RREQ for LAMOR-PA/TR considers the minimum effective node battery power carried in the RREQ. The effective node battery power here is the node’s remaining battery power level minus the transmission power required to deliver $r(\text{RREQ})$ amount to the next hop neighbor. Thus, $b_{eff\_min}(\text{RREQ})$ is updated if the effective node battery power of the upstream neighbor is lower.

If a previous route recorded in the RREQ_Table can carry all the data packets as indicated in $r(\text{RREQ})$, the route carried in the RREQ will be evaluated as in step (f), if the latter route can carry all the data packets as well. Otherwise, the RREQ will be dropped at this stage. On the other hand, if the previous route recorded in the RREQ_Table cannot carry all the data packets as indicated in $r(\text{RREQ})$, either (1) this
route will be replaced with the route carried in the RREQ if the latter route can carry more data packets, or (2) the route carried in the RREQ will be evaluated as in step (f) if the latter route (carried in the RREQ) can carry the same number of data packets. Otherwise, the RREQ will be dropped at this stage as well. In this manner, if low cost routes cannot be built due to power constraints, routes that can carry maximum amount of data (maximum constrained transmission routes) will be constructed. For \textit{LAMOR-PA/TR} in step (f), the route cost $c(m,s,i)$ for the $m^{th}$ route from source $s$ to current node $i$ for \textit{LAMOR-PA/TR} is given in (5.30). This is based on objective function (5.17). The definitions of the different terms in (5.30) can be found in Table 5.1 and 5.2.

\begin{equation}
LAMOR-PA/TR: \\

c(m,s,i) = \left( \frac{e_{\text{tot}}}{\text{max}_e_{\text{tot}}} \right) - \left( \frac{b_{\text{eff}}}{\text{max}_b_{\text{eff}}} \right)
\end{equation}

\begin{align*}
\text{r}_\text{max}_b &= \frac{b_{\text{tot}}(\text{RREQ})}{\text{transmit \_ power}} \\
r_\text{eff} &= \min(r_\text{max}_b, r_{\text{max}(\text{RREQ})}) \\
\text{if } r_{\text{max}(\text{RREQ}_\text{Table})} &\geq r(\text{RREQ}) \\
\text{then if } r_\text{eff} &\geq r(\text{RREQ}) \\
&\text{ then compare\_route\_cost = TRUE} \\
&\text{else exit processing and drop RREQ} \\
\text{else} \\
&\text{if } r_\text{eff} > r_{\text{max}(\text{RREQ}_\text{Table})} \\
&\text{ then } \\
&\text{r}_{\text{max}}(\text{RREQ}) = r_\text{eff} \\
&\text{ compare\_route\_cost = FALSE} \\
&\text{else if } r_\text{eff} = r_{\text{max}(\text{RREQ}_\text{Table})} \\
&\text{ then compare\_route\_cost = TRUE} \\
&\text{else exit processing and drop RREQ} \\
\text{update in RREQ}_\text{Table: } \text{max}_e_{\text{tot}}, \text{max}_b_{\text{eff}}\text{\_min} \text{ and } r_{\text{max}} \\
\text{return compare\_route\_cost}
\end{align*}

Figure 5.7. Check Constraints Module for LAMOR-PA/TR
Chapter 5 Location-aided Multi-Objective Routing

The $max_e_{tot}$, and $max_b_{eff_{min}}$ fields in the RREQ_Table of the node record maximum total transmission power and maximum value of the minimum effective node battery power in a node respectively. These fields are updated each time a RREQ is encountered. Steps (e) and (f) here not only filter and control the flooding of RREQ, but also assist in building up the maximum constrained transmission (in the case of LAMOR-PA/TR) and low cost partial routes respectively till the current forwarding node. These routes are selected for possible route building by the nodes downstream towards the destination node. This also reduces the processing load of the destination node as then it does not need to process an overwhelming number of routes carried by the RREQs.

In the final stage, the values of the $e_{tot}$, $b_{eff_{min}}$ and $r_{max}$ fields in the RREQ_Table will be replaced by the corresponding values obtained from the RREQ. In addition, the sequence of forwarding nodes from $F_{SEQ}(RREQ)$, which is the set of forwarding nodes between the source node and the current intermediate node, will be copied to the RREQ_Table. If the route cost function of the route carried by the RREQ has a lower value or the RREQ carries a route that allows a higher maximum constrained transmission, the node needs to rebroadcast the RREQ if it is not the destination. Before doing so, it will check its own battery power and update $b_{nb} (RREQ)$ accordingly. The node will also update its location in the field $(x_{nb}, y_{nb})$ of the RREQ. Lastly, it will add its IP address into $F_{SEQ}(RREQ)$. If the node is the destination node, and the RREQ received is the first RREQ it receives from a source in a route discovery cycle, the node will wait for some period of time to collect and evaluate the routes carried in the RREQ before it decides on the final route to use. For LAMOR-PA/TR, the route building process in a destination node is different and is summarized in Figure 5.8.
Chapter 5 Location-aided Multi-Objective Routing

\[ r_{\text{max}} = \frac{b_{ab}(\text{RREQ})}{\text{transmit power}} \]
\[ r_{\text{eff}} = \min(r_{\text{max}}, r_{\text{max}}(\text{RREQ})) \]

\textbf{if} (num\_entries(\text{RREQ}\_\text{Table}) \forall src\_entry = s) = 1
\textbf{then if} \sum_{\forall src\_entry = s} r_{\text{max}}(\text{RREQ}\_\text{Table}) \geq r(\text{RREQ})
\textbf{then}
\text{update in \text{RREQ}\_\text{Table}}:
\[ \text{max}_e_{\text{tot}}, \text{max}_b_{\text{eff}}_{\text{min}}, e_{\text{tot}}, b_{\text{eff}}_{\text{min}}, F_{\text{SEQ}} \]
\text{and} \ r_{\text{max}} = r_{\text{eff}}
\textbf{else} exit processing and drop RREQ
\textbf{else}
\text{add another entry into \text{RREQ}\_\text{Table}}:
\[ \text{max}_e_{\text{tot}}, \text{max}_b_{\text{eff}}_{\text{min}}, e_{\text{tot}}, b_{\text{eff}}_{\text{min}}, F_{\text{SEQ}} \text{ and } r_{\text{max}} = r_{\text{eff}} \]

\textbf{else if} (num\_entries(\text{RREQ}\_\text{Table}) \forall src\_entry = s) > 1
\textbf{then if} r_{\text{eff}} \geq r(\text{RREQ})
\textbf{then}
\text{purge all previous entries pointing to source} s
\text{update in \text{RREQ}\_\text{Table}}:
\[ \text{max}_e_{\text{tot}}, \text{max}_b_{\text{eff}}_{\text{min}}, e_{\text{tot}}, b_{\text{eff}}_{\text{min}}, F_{\text{SEQ}} \text{ and } r_{\text{max}} = r_{\text{eff}} \]
\textbf{else}
\textbf{if} \sum_{\forall src\_entry = s} r_{\text{max}}(\text{RREQ}\_\text{Table}) \geq r(\text{RREQ}) \textbf{then}
\begin{enumerate}
\item Add another entry into \text{RREQ}\_\text{Table}:
\[ \text{max}_e_{\text{tot}}, \text{max}_b_{\text{eff}}_{\text{min}}, e_{\text{tot}}, b_{\text{eff}}_{\text{min}}, F_{\text{SEQ}} \text{ and } r_{\text{max}} = r_{\text{eff}} \]
\item Sort all entries for src\_entry = s, putting the entry with largest \ r_{\text{max}} \text{ at the top of the list.}
\item If two or more entries have the same \ r_{\text{max}} \text{ values, sort them}
\text{according to their route cost values in ascending order.}
\item Then remove one entry at a time, starting from the bottom of
\text{the list. Continue to do so until}
\[ \sum_{\forall src\_entry = s} r_{\text{max}}(\text{RREQ}\_\text{Table}) < r(\text{RREQ}) \]
\item Then add back the latest entry that was removed.
\end{enumerate}
\textbf{else}
\text{add another entry into \text{RREQ}\_\text{Table}}:
\[ \text{max}_e_{\text{tot}}, \text{max}_b_{\text{eff}}_{\text{min}}, e_{\text{tot}}, b_{\text{eff}}_{\text{min}}, F_{\text{SEQ}} \text{ and } r_{\text{max}} = r_{\text{eff}} \]

\textbf{Figure 5.8. Routes Building Module in Destination Node for LAMOR-PA/TR}
When the transmission requirement is known, the destination should be able to provide a set of routes which may be used jointly together (in some combination) even if one single route is not able to fully satisfy the transmission requirement. In that case, if a single route residing in the RREQ_Table cannot carry all the data packets in a connection, the destination node will add another route discovered by the subsequent RREQ. The destination node will continue this until the multiple routes collected can together satisfy the transmission requirement. Subsequently, if the destination gets one or more RREQs further, it can choose a set of routes such that it minimizes the overall number of routes to be used for the given transmission requirement. At any point in time, if the destination finds that the incoming RREQ is able to give it a single route that can carry all the data packets, the destination will purge the multiple routes collected and record only this single route. If the destination node has a single route that can carry all the data packets recorded in the RREQ_Table, it will replace this route only if it can find another single route that can carry all data packets and is of a lower cost.

5.3.3 Route Reply Phase

Once the time set to collect routes expires, the destination will construct a route reply (RREP) packet using the information collected in its RREQ_Table. The RREP has the following structure common to all LAMOR-PA protocols: \((src\_addr, dest\_addr, (x_d, y_d), v_d, t_d^0, F\_SEQ)\). LAMOR-PA/TR requires an additional field \(r\_max\). (The definitions of the various fields are given in Table 5.3.) The destination node will then refer to the \(F\_SEQ\) information to forward the RREP to the next hop neighbor towards the source. The IP address of this neighbor is set in the IP header. For LAMOR-PA/TR,
multiple RREPs will be sent by individual unicast packets if multiple routes are required.

Table 5.3. Definition of the various fields in RREP

<table>
<thead>
<tr>
<th>Fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>src_addr</td>
<td>The address of the source node that the node must now reply as the destination.</td>
</tr>
<tr>
<td>dest_addr</td>
<td>The address of the destination node now initiating the RREP.</td>
</tr>
<tr>
<td>((x_d, y_d))</td>
<td>The current co-ordinates of the destination node.</td>
</tr>
<tr>
<td>(v_d)</td>
<td>The current speed if the destination node is in motion, or the weighted average speed if the destination node is currently motionless.</td>
</tr>
<tr>
<td>(t_p^{\text{eff}})</td>
<td>The effective weighted average pause time.</td>
</tr>
<tr>
<td>(t_o)</td>
<td>Marks the time ((x_d, y_d), v_d) and (t_p^{\text{eff}}) are inserted into the RREP</td>
</tr>
<tr>
<td>F_SEQ</td>
<td>Carries the sequence of forwarding nodes (or the route) between the source node and the destination node; this is obtained from the RREQ_Table.</td>
</tr>
<tr>
<td>r_max</td>
<td>Records the maximum transmission that can be carried by this route. Applies to LAMOR-PA/TR only.</td>
</tr>
</tbody>
</table>

Once an intermediate node intercepts a RREP and identifies itself as a potential forwarding node from the IP header, it will take note of the actual transmission power required to reach the immediate downstream node that transmits this RREP. In the case of LAMOR-PA/TR, \(r_{\text{max}}(\text{RREP})\) will be recorded by the intermediate node so that it will know how much battery resource to reserve for this connection. It will then refer to \(F_{\text{SEQ}}(\text{RREP})\) to obtain the next hop neighbor towards the source. After inserting the IP address of the next hop neighbor in the IP header, it will forward the RREP towards the source. When the source node receives the RREP, it will copy the information from RREP packet into its route table. The route table has the following common fields: \(\{\text{dest_addr}, (x_d, y_d), v_d, t_p^{\text{eff}}, t_o, F_{\text{SEQ}}\}\). The definition of the fields for each route entry is given in Table 5.4. In the case of LAMOR-PA/TR, the source node will also record the maximum transmission allowed on each route. The source will then include \(F_{\text{SEQ}}\) in the data packet as source routing is employed to transmit the data packets to
the destination. For data transmission, each node will transmit using an amount of energy just enough to reach the next hop neighbor. Figure 5.9 summarizes the Route Reply Process.

Table 5.4. Definition of the various fields in Route Table

<table>
<thead>
<tr>
<th>Fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>dest_addr</td>
<td>Address of the destination node $d$ initiating a route reply in response to the route request from the source node.</td>
</tr>
<tr>
<td>$(x_d, y_d)$</td>
<td>Indicates the position of the destination node $d$.</td>
</tr>
<tr>
<td>$v_d$</td>
<td>Current or weighted average speed of the destination node $d$.</td>
</tr>
<tr>
<td>$\bar{tp}_d$</td>
<td>Effective weighted average pause time of the destination node $d$.</td>
</tr>
<tr>
<td>$t_{d0}$</td>
<td>The timestamp that indicates the time the values of $(x_d, y_d)$, $v_d$ and $\bar{tp}_d$ are updated in the RREP.</td>
</tr>
<tr>
<td>$F_{.SEQ}$</td>
<td>The sequence of forwarding nodes between the source and destination node $d$.</td>
</tr>
<tr>
<td>$r_{.max}$</td>
<td>Records the maximum transmission that can be carried by this route. Applies to LAMOR-PA/TR only.</td>
</tr>
</tbody>
</table>

**intermediate $k$:**

**if** $k$ is a potential forwarder

**then**

take note of transmission power required to reach downstream neighbor

take note of $r_{.max}$(RREP) for power reservation purpose (for LAMOR-PA/TR only)

forward the RREP to the next upstream neighbor by referring to $F_{.SEQ}$(RREP)

**else if** $k$ is the source node

**then**

record $(dest\_addr, (x_d, y_d), v_d, \bar{tp}_d, t_{d0}, F_{.SEQ})$ from RREP to route table

record maximum transmission allowed on a route (for LAMOR-PA/TR only)

source route data packets using $F_{.SEQ}$

**Figure 5.9. Route Reply Process**

**5.3.4 Performance Studies in Static Network**

This section considers the performance of the three LAMOR-PA protocols in a static network with results obtained both through simulations and optimization through binary
integer programming. Nodes are located within an area of $3R$ by $3R$, with $N$ nodes being randomly placed in each $R$ by $R$ square metre region. $R$ is the radio range of each node, $N$ is a number within $[2, 10]$. In each 20-second interval, 9 nodes will each select a destination node randomly. Each connection period assumes a value randomly selected within $[5, 15]$ seconds. For all connections, two packets will be generated each second, and the size of each packet is 512B. Each node is allocated 16000 battery units or BUs each. 1 BU is required to transmit a packet to a destination 100m away. The energy consumed during control packet transmission is ignored as this amount of energy is negligible as compared to the amount of energy utilized during actual data transmission. For the \textit{LAMOR-PA} protocols, the timeout period for route collection at the destination should occur before the source triggers another route discovery process. Thus, route collection timeout period is set to be one-quarter of the route discovery timeout period (it is 2 seconds as set by the simulator.)

The performance of \textit{LAMOR-PA/TR}, \textit{LAMOR-PAL} and \textit{LAMOR-PAF} obtained through simulations are compared with their respective optimized versions, \textit{LAMOR-PA/TR}(opt), \textit{LAMOR-PAL}(opt) and \textit{LAMOR-PAF}(opt). The three \textit{LAMOR-PA} protocols are simulated using QUALNET [79]. The results for the optimized versions are obtained through binary integer programming using MATLAB. An important measure of system performance for power-limited nodes will be the \textit{network operational time} as this would be indicative of the time for which the network can actually be used. Here, the network operational time is defined as the time at which $n_{\text{drop}}$ percentage of all the data packets generated till then are dropped (i.e. because these data packets cannot be carried by deactivated intermediate nodes). For $n_{\text{drop}}=1$, Figures 5.10, 5.11 and 5.12 show the network operational time against varying node densities for the heuristic schemes.
Chapter 5 Location-aided Multi-Objective Routing

*LAMOR-PA/TR, LAMOR-PAL* and *LAMOR-PAF* respectively along with the results for their respective optimized versions. In each case, the heuristic scheme performs almost as well as its optimized version. The slight degradation in the performance of the heuristic scheme may be attributed to the fact that, unlike the optimized version, it is constrained to search for forwarding nodes within the forwarding zone and also has to implement a time limit in the collection and eventual selection of routes to the destination; this may lead to choice of somewhat less optimal routes in the heuristic schemes. Another useful measure for comparing systems will be the total number of data packets delivered to the destinations within the network operational time. This exhibits a similar trend as observed earlier in the case of network operational time with the heuristic schemes performing only marginally poorer than their corresponding optimized versions. These results show that the practically implementable heuristic schemes perform almost as well as the optimized versions (which are not really implementable but provide a baseline for comparison as they represent the best that can be achieved in principle under the given conditions).

![Figure 5.10. Comparison of LAMOR-PA/TR(opt) and LAMOR-PA/TR](image-url)

Figure 5.10. Comparison of LAMOR-PA/TR(opt) and LAMOR-PA/TR
Chapter 5 Location-aided Multi-Objective Routing

Next, the heuristic algorithms \texttt{LAMOR-PA/TR}, \texttt{LAMOR-PAL} and \texttt{LAMOR-PAF} are compared with \texttt{HEAP} (version 2) and \texttt{PANDA} (Transmission Power version) through simulations. For varying node densities, Figures 5.13(a) and 5.13(b) respectively show

---

Figure 5.11. Comparison of \texttt{LAMOR-PAL(Opt)} and \texttt{LAMOR-PAL}

Figure 5.12. Comparison of \texttt{LAMOR-PAF(Opt)} and \texttt{LAMOR-PAF}
Chapter 5 Location-aided Multi-Objective Routing

the network operational time and the number of packets carried within this network operational time.

Figure 5.13. Performance Comparison of LAMOR-PA protocols against HEAP and PANDA
Chapter 5 Location-aided Multi-Objective Routing

When the node density is low, all the protocols perform similarly as the number of routes available for selection is small. (In the simulations it is observed that under identical conditions, all the protocols almost always select the same route.) At high node densities, the three LAMOR-PA protocols perform better than HEAP by exploring the forwarding zone to find out the least cost route for transmission. Although HEAP searches the entire area, it will select a route as long as the battery power of each forwarding node involved is above a threshold value, and the transmission power of each forwarding node satisfies a selected transmission power level. In this aspect, LAMOR-PA protocols perform better by actually selecting the least cost route among those that are available. Like the LAMOR-PA protocols, PANDA also examines the collected routes first before selecting the final route for use. However, PANDA gives the lowest network operational time and total number of packets carried within the network operational time. This is because PANDA only considers choosing the route that requires the least transmission power. It never considers the remaining battery power of the forwarding nodes along the route, which determines the network operational time here. In contrast, the LAMOR-PA protocols do consider the remaining battery power of the forwarding nodes and the total transmission power required during route selection.

Among the three LAMOR-PA protocols, LAMOR-PAL does not perform as well as LAMOR-PA/TR and LAMOR-PAF. At high node density of 10 nodes per \( R \) by \( R \) region, LAMOR-PA/TR and LAMOR-PAF each show 10% improvement in terms of network operational time and number of data packets carried with the network operational time, as compared to LAMOR-PAL. LAMOR-PAL considers the maximum load carried or energy consumed in the past interval by the intermediate nodes along a route to decide this route cost. This is not as accurate as considering the current load that is carried by
an intermediate node as in the case of LAMOR-PA/TRA. For LAMOR-PA/TRA, with the current load information given, a more accurate view of the remaining battery power after deducting an amount needed to carry the current load can be obtained. This will allow a node to better select a least cost route. Node lifetime can be extended and this in turn will prolong the network operational time as well. In addition, LAMOR-PA/TRA allows a single or multiple routes to be discovered to carry the entire connection through. For LAMOR-PA/TRA, the amount of data to be transmitted must be declared upfront and is more suitable for file transfer and download applications. LAMOR-PAF performs as well as LAMOR-PA/TRA. This is because the former adds another factor in the route cost function. On top of considering the load experienced so far as reflected by the remaining battery power, the forwarding load is also considered separately. Minimizing the forwarding load of a node in the case of LAMOR-PAF helps to reserve battery power for each node's self transmission. This aids in improving individual node's lifetime as well.

5.4 Power and Mobility Aware Multi-Objective Routing

5.4.1 Problem Formulation

Since node mobility affects the stability and lifetime of a route going through it, this needs to be considered as an additional objective in a mobile environment. When a forwarding node moves out of range, the associated links will break. A route that comprises such a link can no longer be used and data packets along such a route will be dropped. A source node will have to rediscover another route again if required. Therefore, route lifetime, or the period of time a route can remain intact to carry data traffic, is an important consideration. Route lifetime is dependent on the minimum link
lifetime which would need to be derived. Consider the scenario of Figure 5.14. Given that node $i$ and node $j$ are originally located at $\{x_i, y_i\}$ and $\{x_j, y_j\}$ respectively, separated by a distance $D < R$ where $R$ is the radio range, $v_i$ and $v_j$ are the speeds, and $\theta_i$ and $\theta_j$ are the moving directions of nodes $i$ and $j$, respectively. These will move to positions $\{x_i', y_i'\}$ and $\{x_j', y_j'\}$ respectively before they get separated by a distance greater than the radio range and are disconnected. By then, they would have moved a distance of $d_i$ and $d_j$ respectively. Therefore, link lifetime is given by:

$$t = \frac{d_i}{v_i} = \frac{d_j}{v_j} \quad (5.31)$$

Applying cosine rule:

$$R^2 = d_i^2 + (D + d_j)^2 - 2d_i(D + d_j)\cos(\theta_i - \theta_j) \quad (5.32)$$

![Diagram for Link Lifetime Derivation](image)

Figure 5.14. Diagram for Link Lifetime Derivation

Substituting $d_j = \frac{d_i}{v_i} v_j$ into (5.32) and rearranging, a quadratic equation which can be solved for $d_i$ will be obtained. The link lifetime $t$ will then be given by:

$$t = -Dz_1 + \sqrt{R^2(z_3) - (Dz_2)^2} \quad \frac{z_3}{z_3} \quad (5.33)$$
Chapter 5 Location-aided Multi-Objective Routing

where \( z_1 = v_j - v_i \cos(\theta_i - \theta_j) \) \( \text{(5.34)} \)

\[ z_2 = v_i \sin(\theta_i - \theta_j) \] \( \text{(5.35)} \)

\[ z_3 = v_i^2 + v_j^2 - 2v_i v_j \cos(\theta_i - \theta_j) \] \( \text{(5.36)} \)

\[ D = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \] \( \text{(5.37)} \)

Actually, there are two solutions, \( t = \frac{-Dz_1 + \sqrt{R^2(z_3) - (Dz_2)^2}}{z_3} \) and

\[ t = \frac{-Dz_1 - \sqrt{R^2(z_3) - (Dz_2)^2}}{z_3} \], when the quadratic equation is solved.

Note that \( z_3 = v_i^2 + v_j^2 - 2v_i v_j \cos(\theta_i - \theta_j) = (v_i \cos \theta_i - v_j \cos \theta_j)^2 + (v_i \sin \theta_i - v_j \sin \theta_j)^2 \),

so \( z_3 \) is always positive. If \(-Dz_1 + \sqrt{R^2(z_3) - (Dz_2)^2} > 0\) and

\(-Dz_1 - \sqrt{R^2(z_3) - (Dz_2)^2} < 0\) can be proven, then it can be shown that there is only one possible positive solution; the other being a negative solution.

Squaring both sides and re-arranging \(-Dz_1 + \sqrt{R^2(z_3) - (Dz_2)^2} > 0\) gives \( R^2z_3 > D^2z_3 \)

which is valid, since \( R > D \). Therefore, \( t = \frac{-Dz_1 + \sqrt{R^2(z_3) - (Dz_2)^2}}{z_3} \) will always give a positive solution. Following the same line of reasoning, \( t = \frac{-Dz_1 - \sqrt{R^2(z_3) - (Dz_2)^2}}{z_3} \)

will always give a negative solution. Therefore, it is shown that there is only one possible positive solution, or only one possible solution because the other solution gives
Chapter 5 Location-aided Multi-Objective Routing

a negative value when \( D < R \); link lifetime \( t \) can never assume a negative value. Another formulation for the link lifetime, using the same parameters, can be found in [88]-[89].

Three variations for a power and mobility aware multi-objective problem for mobile networks are proposed. The first variation states (5.38) as the objective to be satisfied and uses (5.39)-(5.45) as the associated constraints by incorporating mobility awareness in the multi-objective formulation of Section 5.2.1. In the first additional constraint (5.41), \( t_g^m \) is the time node \( g \) stays connected to the next downstream node along the \( m \)th route, i.e. this is the link lifetime between node \( g \) and the next downstream node. The route lifetime \( t_{W}^m \) of the \( m \)th route is determined by the lifetime of the weakest link. The second additional constraint (5.43) states that the route to be selected must still be intact. The additional objective \( \max(\sum_{V^m} t_{W}^m f_{m}^m) \) that maximizes the route lifetime has to be considered for route selection here. Since \( \max(\sum_{V^m} t_{W}^m f_{m}^m) \) is considered to be as important as the other objective functions, a value of 1 is assigned to its associated weight as well. \( \max(\sum_{V^m} t_{W}^m f_{m}^m) \) is normalized by the maximum weakest link lifetime \( t_{W_{max}} \). The objective here is to choose a route based on minimizing the total transmission energy required, maximizing the minimum remaining battery energy and maximizing the route lifetime as in (5.38).

\[
\min \left( \sum_{V^m} e_{m}^m f_{m}^m \right) - \left( \sum_{V^m} b_{m}^m f_{m}^m \right) - \left( \sum_{V^m} t_{W}^m f_{m}^m \right)
\]

subject to:

\[
b_{m}^m = \min_{v \in F^m} b_{v}^m \quad \forall m
\]

\[
e_{m}^m = \sum_{v \in F^m} e_{v}^m \quad \forall m
\]
Chapter 5 Location-aided Multi-Objective Routing

\[ tw^m = \min_{\forall g \in F^m} t^m \]  \hspace{1cm} (5.41)

\[ b^m f^m > 0 \ \forall m \]  \hspace{1cm} (5.42)

\[ tw^m f^m > 0 \ \forall m \]  \hspace{1cm} (5.43)

\[ \sum_{\forall m} f^m = 1 \]  \hspace{1cm} (5.44)

\[ f^m = [0,1] \ \forall m \]  \hspace{1cm} (5.45)

For the second variation, only power awareness is considered in the objective function, i.e. this variation consists of objective function (5.10) and the associated constraints (5.39) to (5.45).

5.4.2 Variation with Known Transmission Requirement

The third variation considers \( r \), the total data size to be transmitted, and \( tc \), the entire communication session time, to be given as considered earlier in Section 5.2.3 but adds mobility awareness into the power aware multi-objective formulation given for that case. This implies that the third variation of the power and mobility aware multi-objective problem comprises objective function (5.17) with associated constraints (5.18) to (5.23). In addition, two additional constraints, (5.41) as given earlier and (5.46) are included. Constraint (5.46) implies that the route selected must be able to support the entire communication session of time duration \( tc \).

\[ tw^m f^m \geq tc \ \forall m \]  \hspace{1cm} (5.46)

If both battery power constraint (5.21) and route lifetime constraint (5.46) are satisfied, the least cost route (or the route with the minimum objective function value) will be
selected. This will give a solution when the multi-objective problem is solved. However, if either or both of the constraints are not satisfied, no solution can be found. In this case, the strategy here will be to first find if there are any routes that satisfy (5.21) but not (5.46). If there are, the route that allows maximum transmission, or the route that has the maximum lifetime, will be selected among these routes. The primary aim is not to drain out a node’s battery capacity as far as possible. However, if there are no such routes, then this implies that the routes to be evaluated in this case either violate both constraints or only violate constraint (5.21). The following steps will have to be followed. Assume that the amount of data that can be transmitted along the \( m \text{th} \) route for a duration of the route lifetime \( t \omega^m \), which is denoted as \( r(t \omega^m) \) is known. The effective amount of data that can be transmitted as constrained by battery power needs to be computed too. This is given by \( r_{\text{eff}}^m = \min_{v \in \mathcal{P}} \left( \frac{b_v - e_{\text{eg}}^m}{e_{\text{eg}}^m} \right) \). The actual amount of data that can be transmitted along the \( m \text{th} \) route will be given by \( r_{\text{act}}^m = \min( r_{\text{eff}}^m, r(t \omega^m) ) \). Route selection will then be based on selecting the \( m \text{th} \) route based on \( \max_{v \in \mathcal{P}} r_{\text{act}}^m \). This process needs to be repeated again to find additional routes which can be used to transmit the remaining data packets.

5.5 On-demand Location-aided Power and Mobility Aware Routing

5.5.1 Route Discovery

As in the case of static networks (Section 5.3), three heuristic location-aided multi-objective routing protocols which are both power-aware and mobility-aware are proposed for use in mobile networks. These are schemes with a mobility aware objective (LAMOR-PMAMO), with route lifetime information (LAMOR-PMALF) and
with transmission requirement given (LAMOR-PMA/TR), corresponding to the three variations formulated in Sections 5.4.1 and 5.4.2. Since all these three LAMOR-PMA protocols require route lifetime information, the minimum link lifetime has to be determined first using (5.33) to (5.37). This requires a node to know its location, speed and heading direction. As each node may not be equipped with a motion sensing device, a simple motion prediction method is proposed for a node to obtain its speed and heading direction. Given that only the location information is available, the strategy is to obtain the speed and heading direction by tracking its location every $\Delta t_{\text{track}}$ seconds since the start of its operation. If node $i$ finds that its current position $(x_i(t), y_i(t))$ at time $t$ is the same as that $\Delta t_{\text{track}}$ seconds earlier, as in $x_i(t) = x_i(t - \Delta t_{\text{track}})$ and $y_i(t) = y_i(t - \Delta t_{\text{track}})$, then it assumes that it is stationary at the current time. However, if it finds that its position has changed, it will take note of the start time $t_{\text{start}}$ and start position $(x_i(t_{\text{start}}), y_i(t_{\text{start}}))$. As long as the node finds that its position has changed, it will calculate its current speed $v_i(t)$ and heading direction $\theta_i(t)$ based on the start time and start position. The motion prediction module is summarized in Figure 5.15.

The route discovery process for these three protocols is similar to that described for the three LAMOR-PA protocols of Section 5.3. However, in the route request phase of the three LAMOR-PMA protocols, the RREQ initiated contains the additional fields as indicated in Table 5.5. The additional fields in the RREQ_Table are indicated in Table 5.6.
<table>
<thead>
<tr>
<th>If $x_i(t) = x_i(t - \Delta t_{\text{track}})$ and $y_i(t) = y_i(t - \Delta t_{\text{track}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_i(t) = 0$</td>
</tr>
<tr>
<td>$\theta_i(t) = 0$</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>$d_x = (x_i(t) - x_i(t_{\text{start}}))$</td>
</tr>
<tr>
<td>$d_y = (y_i(t) - y_i(t_{\text{start}}))$</td>
</tr>
<tr>
<td>$v_i(t) = \frac{\sqrt{d_x^2 + d_y^2}}{t - t_{\text{start}}}$</td>
</tr>
<tr>
<td>If $x_i(t) &gt; x_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>If $y_i(t) &gt; y_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>$\theta_i(t) = \arctan(d_y / d_x)$</td>
</tr>
<tr>
<td>Else if $y_i(t) = y_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>$\theta_i(t) = 0$</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>$\theta_i(t) = 2\pi - \arctan(d_y / d_x)$</td>
</tr>
<tr>
<td>Else if $x_i(t) = x_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>If $y_i(t) &gt; y_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>$\theta_i(t) = \pi / 2$</td>
</tr>
<tr>
<td>Else if $y_i(t) &lt; y_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>$\theta_i(t) = 3\pi / 2$</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>If $y_i(t) &gt; y_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>$\theta_i(t) = \pi - \arctan(d_y / d_x)$</td>
</tr>
<tr>
<td>Else if $y_i(t) = y_i(t_{\text{start}})$</td>
</tr>
<tr>
<td>$\theta_i(t) = \pi$</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>$\theta_i(t) = \pi + \arctan(d_y / d_x)$</td>
</tr>
</tbody>
</table>

Figure 5.15. Motion Prediction Module
Table 5.5. Definition of the additional fields in RREQ for LAMOR-PMA protocols

<table>
<thead>
<tr>
<th>Fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{ab}$</td>
<td>Records the speed of immediate upstream neighbor. The source node will initialize with its own speed.</td>
</tr>
<tr>
<td>$\theta_{ab}$</td>
<td>Records the heading direction of immediate upstream neighbor. The source node will initialize with its heading direction.</td>
</tr>
<tr>
<td>route_lifetime</td>
<td>The lifetime of a route as the RREQ propagates from the source to the destination node. This is initialized to zero by the source node.</td>
</tr>
<tr>
<td>$r$</td>
<td>Records the number of data packets the source will transmit in this connection to be made. Available in LAMOR-PMA/TR only.</td>
</tr>
<tr>
<td>$r_{max}$</td>
<td>The maximum number of data packets that can be transmitted along the route built. The source node will initialize with a value of zero. Available in LAMOR-PMA/TR only.</td>
</tr>
<tr>
<td>$b_{ab}$</td>
<td>Records the remaining battery power of the immediate upstream neighbor. It is the current remaining battery power level less the amount reserved for the ongoing transmissions. The source node will initialize with its remaining battery power. Available in LAMOR-PMA/TR only.</td>
</tr>
<tr>
<td>b_eff_min</td>
<td>Records the minimum effective node battery power as the RREQ propagates from the source to the destination node. The effective node battery power here is the node’s remaining battery power level minus the transmission power required to deliver $r$ amount of data to the next hop neighbor. This is initialized to zero by the source. Applies to LAMOR-PMA/TR only.</td>
</tr>
<tr>
<td>t_connect</td>
<td>Records the connection time of this connection to be made. Applies to LAMOR-PMA/TR only.</td>
</tr>
</tbody>
</table>

Table 5.6. Definition of the additional fields in RREQ_Table for LAMOR-PMA protocols

<table>
<thead>
<tr>
<th>Fields</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>route_lifetime</td>
<td>The lifetime of a route between the source node $s$ and the node that records the route request information.</td>
</tr>
<tr>
<td>max_route_lifetime</td>
<td>The maximum route lifetime found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$. Applies to LAMOR-PMAMO only.</td>
</tr>
<tr>
<td>$r_{max}$</td>
<td>The maximum number of data packets that can be transmitted along the route built and seen so far, pertaining to node $s$. Applies to LAMOR-PMA/TR only.</td>
</tr>
<tr>
<td>b_eff_min</td>
<td>The minimum effective node battery power found on a route between the source node and the node that records the route request information. Available in LAMOR-PMA/TR only.</td>
</tr>
<tr>
<td>max_b_eff_min</td>
<td>The maximum value of the minimum effective node battery power found among all the routes that the node has seen from the RREQs propagated and pertaining to node $s$. Available in LAMOR-PMA/TR only.</td>
</tr>
</tbody>
</table>
Chapter 5 Location-aided Multi-Objective Routing

Figure 5.16 and 5.17 sum up the Route Request Process at intermediate nodes for LAMOR-PMAMO, LAMOR-PMALF and LAMOR-PMA/TR respectively. When an intermediate node intercepts a RREQ, the LAMOR-PMA protocols require the following additional steps. Using the location, speed and heading direction of the immediate upstream node carried in the RREQ and its own corresponding parameters, the lifetime of the link between these two nodes can be determined. If the computed link lifetime is lower than the route\_lifetime(RREQ), the former will be set as the updated route\_lifetime(RREQ). If the node finds that the updated route\_lifetime(RREQ) value has expired, the RREQ will not be processed anymore and is dropped immediately, as a link along this route has broken. Similarly, any entry in the RREQ\_Table that has an expired route lifetime will also be removed. In addition, for LAMOR-PMA/TR the node will also proceed to check the battery power and route lifetime constraints, as summarized in Figure 5.18, before evaluating the route cost. A fixed transmission rate transmit\_rate is assumed here but a variable transmission rate may also be carried by the RREQ. The maximum amount of data packets r\_max\_b and r\_max\_tc the upstream neighbor can send subject to battery power and route lifetime constraints respectively can be computed as shown in Figure 5.18. Once these have been computed, the effective maximum amount of data packets r\_eff that the route in the RREQ can carry can then be computed as reflected in Figure 5.18.

If a previous route recorded in the RREQ\_Table can carry all the data packets as indicated in r(RREQ), the route cost carried in the RREQ will be evaluated if the latter route can carry all the data packets as well. Otherwise, the RREQ will be dropped at this stage. If the previous route recorded in the RREQ\_Table cannot carry all the data packets as indicated in r(RREQ), this could be due to the violation of battery power or
Chapter 5 Location-aided Multi-Objective Routing

route lifetime constraints or both. If this is due to the route lifetime constraint violation alone, then the route carried in the RREQ will still be considered if (i) the latter route satisfies both the battery power and route lifetime constraints, or (ii) the latter route satisfies the battery power constraint only but has a higher route lifetime. If this is due to battery power constraint violation only or violations of both constraints, the route carried in the RREQ will be considered if (i) the latter route satisfies battery power constraint only, or (ii) effective maximum amount of data packets $r_{eff}$ that the route in the RREQ can carry is higher. The primary aim here is not to drain out a node’s battery capacity as far as possible (as explained in Section 5.4.2), which is why the battery power constraint will be satisfied as far as possible. In this case, a route that does not satisfy the route lifetime constraint may be selected. However, the source node can then perform another route discovery to find another route that can complete the source transmission and not completely drain out any of the forwarding nodes. This would be particularly effective in mobile situations as, over time, some other nodes may move in range which may then be used to create new routes. For a route in the RREQ that is being considered as a result of this constraint checking process, it will be updated in the RREQ_Table, and the RREQ will be broadcast if the node concerned is an intermediate node.

For the route in the RREQ that needs to be evaluated based on its route cost, the route cost for \textit{LAMOR-PMAMO} and \textit{LAMOR-PMALF} will be (5.47) and (5.48) respectively; the route cost for \textit{LAMOR-PMA/TR} will be (5.30). If the route cost of the route carried by the RREQ is lower, the route will be updated in the RREQ_Table, and the RREQ will be broadcast if the node concerned is an intermediate node. The intermediate node
will update the additional fields $v_{nb}$ and $\theta_{ab}$ as well in the RREQ so the next downstream node can compute link and route lifetime.

**LAMOR-PMAMO:**

\[ c(m, s, i) = \left( \frac{e_{tot}}{\max_e_{tot}} \right) - \left( \frac{b_{\min}}{\max_b_{\min}} \right) - \left( \frac{\text{route\_lifetime}}{\max_{\text{route\_lifetime}}} \right) \]  \hspace{1cm} (5.47)

**LAMOR-PMALF:**

\[ c(m, s, i) = \left( \frac{e_{tot}}{\max_e_{tot}} \right) - \left( \frac{b_{\min}}{\max_b_{\min}} \right) \]  \hspace{1cm} (5.48)

The RREP for all *LAMOR-PMA* protocols contain an additional *route\_lifetime* field. The source node will need this information so it can pre-empt a link breakage and discover a new route again before the link actually breaks. This will prevent data packets from being dropped during an ongoing communication session. For *LAMOR-PMA/TR*, the RREP will also contain the *r\_max* field to indicate the maximum amount of data packets that can be carried and the \((x_{nb}, y_{nb})\) field to indicate the previous downstream node position. These fields will enable an intermediate node, upon receiving a RREP, to know how much battery power resource to reserve for the route carried by the RREP. However, it should be noted that since the nodes are mobile, the committed battery power may not actually be used when the intermediate node moves away breaking its connection. In this case, a timeout mechanism will need to be incorporated that will trigger the transfer of the committed battery power back to the available battery power pool. The transfer will begin when it is found that the committed battery power has not been used even though the timeout period has expired.
Chapter 5 Location-aided Multi-Objective Routing

intermediate \( k \):

\[
\text{if } (\text{src_addr, sequence_no}) \text{ of RREQ is the same OR more updated}
\]

\[
\text{then}
\]

\[
\text{CHK_ROUTE_FN = FALSE}
\]

\[
\text{if } k \text{ is the destination } d
\]

\[
\text{then} \text{ CHK_ROUTE_FN = TRUE}
\]

\[
\text{else if } k \text{ is NOT the source } s \text{ OR destination } d
\]

\[
\text{then if } k \text{ is in F_ZONE(RREQ)}
\]

\[
\text{then if } k \text{ is nearer to } d \text{ than neighbor } nb
\]

\[
\text{then} \text{ CHK_ROUTE_FN = TRUE}
\]

\[
\text{if } (\text{CHK_ROUTE_FN = TRUE})
\]

\[
\text{then}
\]

\[
\text{update } e_{tot}(RREQ), \text{ route_lifetime}(RREQ)
\]

\[
\text{if } \text{route_lifetime}(RREQ) \text{ has expired}
\]

\[
\text{then} \text{ exit processing and drop RREQ}
\]

\[
\text{if } \text{route_lifetime}(RREQ_{\text{Table}}) \text{ has expired}
\]

\[
\text{then remove entry}
\]

\[
\text{update in RREQ_{Table}: max_e_tot, max_b_min}
\]

\[
\text{and max_route_lifetime(LAMOR-PMAMO only)}
\]

\[
\text{if } c(m, s, k) \text{ (RREQ)} \geq c(m', s, k) \text{ (RREQ_{Table})}
\]

\[
\text{then} \text{ exit processing and drop RREQ}
\]

\[
\text{update in RREQ_{Table} from RREQ: e_{tot}, b_{min},}
\]

\[
\text{route_lifetime, F_SEQ}
\]

\[
\text{if } k \text{ is NOT the destination } d
\]

\[
\text{then}
\]

\[
\text{update in RREQ from node: b_{min}, b_{nb}, (x_{nb}, y_{nb}),}
\]

\[
v_{nb}, \theta_{nb}, \text{F_SEQ}
\]

\[
\text{rebroadcast RREQ}
\]

\[
\text{else if } k \text{ is the destination } d
\]

\[
\text{then if } (\text{src_addr, sequence_no}) \text{ of RREQ is more updated}
\]

\[
\text{then} \text{ set timer to collect routes}
\]

Figure 5.16. Route Request Process for LAMOR-PMAMO and LAMOR-PMALF
intermediate $k$:

if $(src\_addr, sequence\_no)$ of RREQ is the same OR more updated
then
  CHK_ROUTE\_FN = FALSE

  if $k$ is the destination $d$
    then CHK_ROUTE\_FN = TRUE
  else if $k$ is NOT the source $s$ OR destination $d$
    then if $k$ is in $F\_ZONE$(RREQ)
      then if $k$ is nearer to $d$ than neighbor $nb$
        then CHK_ROUTE\_FN = TRUE

if (CHK_ROUTE\_FN = TRUE)
then
  update $b\_eff\_min$(RREQ), $e\_tot$(RREQ) and $route\_lifetime$(RREQ)

  if $route\_lifetime$(RREQ) has expired
    then exit processing and drop RREQ

  if $route\_lifetime$(RREQ\_Table) has expired
    then remove entry

if (check\_constraints\_module) (See Figure 5.18) = TRUE
then if $c(m, s, k)$ (RREQ) $\geq c(m', s, k)$ (RREQ\_Table)
  then exit processing and drop RREQ

update in RREQ\_Table from RREQ: $e\_tot, b\_eff\_min, route\_lifetime, F\_SEQ$

if $k$ is NOT the destination $d$
then
  update in RREQ from node: $b_{nb}$, $(x_{nb}, y_{nb})$, $v_{nb}$, $\theta_{nb}$, $F\_SEQ$
  rebroadcast RREQ
else if $k$ is the destination $d$
then if $(src\_addr, sequence\_no)$ of RREQ is more updated
  then set timer to collect routes

Figure 5.17. Route Request Process for LAMOR-PMA/TR
Chapter 5 Location-aided Multi-Objective Routing

\[ r_{\text{max}} = \frac{b_{\text{rb}}(RREQ)}{\text{transmit power}} \]

\[ r_{\text{max}} = \text{transmit rate} \times \text{route lifetime}(RREQ) \]

\[ r_{\text{eff}} = \min(r_{\text{max}}b, r_{\text{max}}tc, r_{\text{max}}(RREQ)) \]

if \( r_{\text{max}}(RREQ) \geq r(RREQ) \)

then if \( r_{\text{eff}} \geq r(RREQ) \)

then \( \text{compare_route_cost} = \text{TRUE} \)

else exit processing and drop RREQ

else if \( b_{\text{eff min}}(RREQ) \geq 0 \)

then if \( b_{\text{eff min}}(RREQ) \geq 0 \) AND \( \text{route lifetime}(RREQ) \geq t_{\text{connect}}(RREQ) \)

then \( r_{\text{max}}(RREQ) = r_{\text{eff}} \)

\( \text{compare_route_cost} = \text{FALSE} \)

else if \( b_{\text{eff min}}(RREQ) \geq 0 \) AND \( \text{route lifetime}(RREQ) < t_{\text{connect}}(RREQ) \)

then if \( \text{route lifetime}(RREQ) > \text{route lifetime}(RREQ)_{\text{Table}} \)

then \( r_{\text{max}}(RREQ) = r_{\text{eff}} \)

\( \text{compare_route_cost} = \text{FALSE} \)

else exit processing and drop RREQ

else exit processing and drop RREQ

else if \( b_{\text{eff min}}(RREQ) \geq 0 \)

then \( r_{\text{max}}(RREQ) = r_{\text{eff}} \)

\( \text{compare_route_cost} = \text{FALSE} \)

else if \( r_{\text{eff}} > r_{\text{max}}(RREQ)_{\text{Table}} \)

then \( r_{\text{max}}(RREQ) = r_{\text{eff}} \)

\( \text{compare_route_cost} = \text{FALSE} \)

else exit processing and drop RREQ

update in RREQ_Table: max_route_lifetime, max_e_tot, max_b_eff_min and \( r_{\text{max}} \)

return \( \text{compare_route_cost} \)

Fig 5.18. Check Constraints Module for LAMOR-PMA/TR
5.5.2 Route Maintenance

If a source node finds that a route lifetime has expired, it will initiate another route discovery process if data transmission to the affected destination is still required. This will reduce the amount of route error packets initiated and the end-to-end delay as well. As speed and heading direction are estimated, and motion status of forwarding nodes may change during data transmission, some link breakages cannot be pre-empted. In this case, once a forwarding node detects a link break and receives a data packet that needs to be sent along the affected link, it will construct a route error packet. Link breakages can be detected via medium sensing at the MAC layer. The route error packet (RERR) will include the extracted $F_SEQ$ from the data packet, and is then transmitted upstream to the next hop neighbor towards the source. Once an intermediate node receives the RERR and it recognizes itself as the rightful forwarding node, it will forward the RERR upstream by referring to $F_SEQ$. When the RERR reaches the source node, the source node will then perform another route discovery, if that is still required.

5.5.3 Performance Evaluation in Mobile Network

The performances of the three heuristic $LAMOR-PMA$ schemes are studied in mobile networks. For the mobile scenarios here, each node will move in a random waypoint manner where each node will move with a speed randomly chosen within [1, 20] m/s with a pause time $P$ seconds in between movements. $P$ is varied within the range [30, 300] seconds for the performance evaluation here. The node density is kept at 10 nodes per $R$ by $R$ square metre. The optimal schemes will not be simulated here since it is infeasible to deploy the optimized versions of $LAMOR-PMA$ in mobile situations.
Figure 5.19 shows the plots of the network operational time against the varying pause
times for HEAP, PANDA, LAMOR-PMA/TR, LAMOR-PMAMO and LAMOR-PMALF.
\( n_{\text{drop}}=2 \) is set here as node mobility may result in more data packets being dropped. It is
observed that with route lifetime information given, all LAMOR-PMA protocols enable
nodes to pre-empt link breakages so that nodes can discover routes again before data
packets are dropped at the intermediate nodes. This results in longer network
operational times for all the LAMOR-PMA protocols. LAMOR-PMALF may seem to be
a better performer here as compared to LAMOR-PMAMO. However, in the case of
LAMOR-PMAMO, the end of the network operational time occurs after the end of the
network lifetime. Network lifetime is defined as the time till the first node has depleted
its battery power resource. This means that within the network lifetime, less than 2% of
the total data packets generated are dropped. This is because LAMOR-PMAMO ensures
that route lifetime is considered as well while choosing routes, resulting in more stable
routes to be selected. After the end of the network lifetime, nodes start to deplete their
battery power one by one. This will then increase the packet dropping rate. When node
mobility is high, i.e. for \( P=30 \)s, LAMOR-PMAMO performs better. In the case of
LAMOR-PMALF, the end of the network operational time occurs before the end of the
network lifetime. Among all the LAMOR-PMA protocols, LAMOR-PMA/TR performs
the best. The route cost function of LAMOR-PMA/TR only considers power awareness
(transmission power and battery power), as in the case of LAMOR-PMALF, thereby
extending the network lifetime. LAMOR-PMA/TR does not include route lifetime in the
route cost function, which otherwise may cause a node not to choose the most energy
efficient route. However, it considers the routes for route cost evaluation only if those
routes satisfy both battery power and route lifetime constraints \((5.21) \) and \((5.46) \)
respectively). Thus, it is able to select stable routes in this manner and the end of the
network operational time occurs after the end of the network lifetime. It is observed that when node mobility is high, i.e. for $P=30s$, *LAMOR-PMA/TR* outperforms all the other protocols.

![Graph of Network Operational Time against Pause Time](image)

Figure 5.19. Graph of Network Operational Time against Pause Time

To evaluate the energy efficiency of the various protocols, their network lifetime performances need to be studied. Figure 5.20 and 5.21 show the plots of the network lifetime and the number of packets carried within the network lifetime respectively against the varying values of the pause time for the five protocols. From the results shown, as mobility increases (i.e. pause time decreases), the network lifetime and the corresponding number of packets carried increase for all protocols. This is because higher mobility allows neighbors, and the distances between nodes, to be changed frequently because of node movements. This allows lower-energy forwarding nodes to be replaced by higher-energy nodes so there is a better overall distribution of energy usage for forwarding. A node here will not need to stay in a position too long to serve as
Chapter 5 Location-aided Multi-Objective Routing

a forwarding node, especially in situations where it is located along one of the few routes between a source-destination pair.

Figure 5.20. Graph of Network Lifetime against Pause Time

Figure 5.21. Graph of Number of Data Packets Carried within Network Lifetime against Pause Time
Figure 5.20 and 5.21 show that both LAMOR-PMM/TR and LAMOR-PMALF perform better than LAMOR-PMAMO, in terms of network lifetime and throughput performance. This is because the former two LAMOR-PMA protocols consider power awareness only when selecting routes. LAMOR-PMAMO may not choose the most energy efficient route since it also needs to consider route lifetime or stability. The performance of PANDA improves in mobile situations as here also mobility helps to remove a low-powered node from a route. As shown in Fig. 5.20 and 5.21, the network lifetime and throughput performances of PANDA and LAMOR-PMAMO are comparable. LAMOR-PMA/TR and LAMOR-PMALF perform better since they place equal importance on total transmission power and remaining node battery power. As all protocols, except HEAP, are more effective in selecting the least cost route, a difference in forwarding node positions will reflect a corresponding difference in route cost, which affects a node’s decision on which route to select. For HEAP, the positions of two forwarding nodes may not affect its decision making if transmitting to both nodes satisfy the selected transmission power level. As a result, improvement in the performance of HEAP is not as significant as compared to the improvement shown by the other schemes as pause time increases.

Figure 5.22 shows the plot of the energy usage against the pause time for the five protocols. The performance of PANDA, LAMOR-PMA/TR and LAMOR-PMALF are comparable, because their route selection is based on power awareness factors. With the inclusion of route lifetime in the route cost, LAMOR-PMAMO may not always choose the most energy efficient route. As PANDA and all the LAMOR-PMA protocols are more effective in selecting the least cost route, this creates a large performance gap between the former four protocols and HEAP. Increasing mobility increases the chance
Chapter 5 Location-aided Multi-Objective Routing

of bringing in energy-rich intermediate nodes, and relocating nodes closer to each other, resulting in lower amount of transmission energy required for all routing protocols as shown in Figure 5.22. As mobility increases, link breakages occur more often and this leads to a higher number of RREQs generated to search for routes again. However, the ratio of the size of a RREQ packet to that of a data packet is about 1:10, so the amount of energy required to transmit a RREQ packet is very much lower as compared to the amount required to transmit a data packet. The increase in the amount of transmission energy required to transmit the additional RREQ packets is not as significant as the reduction in the total amount of transmission energy required to transmit the data packets. Thus, increasing mobility will lead to energy savings for the nodes.

Figure 5.22. Graph of Energy Usage against Pause Time

Figure 5.23 shows the graph of the number of route error packets being generated per connection against the various pause time values. From Figure 5.23, it is shown that all the LAMOR-PMA protocols perform much better than HEAP and PANDA in this case. Route error packet generation is caused by link breakages when there are ongoing data transmissions via the affected links. With the knowledge of route lifetime incorporated
in the *LAMOR-PMA* protocols, a source node can pre-empt link breakage on an existing route and initiate another route discovery process even before the link actually breaks. Therefore, route error packet generation rate is not very high in the cases of the *LAMOR-PMA* protocols. For all the *LAMOR-PMA* protocols, some route error packets may still be generated because node mobility is estimated based only on its past few positions. There will be cases where a node may start moving when it is predicted to be motionless, or where a node may stop moving when it is predicted to be still in motion. However, Figure 5.23 shows that the proposed prediction method is nevertheless able to significantly lower the amount of route error packets generated.

![Figure 5.23. Graph of Average Number of Route Error Packets (RERR) generated per Connection against Pause Time](image)

If route lifetime or stability is not considered during route selection, as in the case of *LAMOR-PMALF*, the average end-to-end delay will be 30%-80% higher compared to the delay incurred in the case of *LAMOR-PMAMO*. This is because nodes in *LAMOR-PMALF* encounter link breakages more frequently. This means that nodes here have to spend some time discovering routes again within the data transmission period.
average end-to-end delay of *LAMOR-PMA/TR* is close to that of *LAMOR-PMAMO*. This is because *LAMOR-PMA/TR* is also capable of selecting stable routes with the help of its constraint checking module as that checks route lifetime constraint for route selection. It is observed that the average end-to-end delay in the case of *PANDA* can be two to five times higher than that of *LAMOR-PMALF*. For *PANDA*, the delay includes the deferring period required to rebroadcast route request packets, on top of the delay incurred due to the route discovery required when links break during data transmission. The average end-to-end delay of *HEAP* is the lowest since RREQ rebroadcast is immediate (there is no deferring period involved) and RREP dissemination is also instantaneous (there is no waiting time for route collection at the destination).

Figure 5.24 shows the graph of the ratio of the number of RREQs forwarded to the number of RREQs initiated against pause time for all the five protocols. Despite having to look for more routes to evaluate and obtain the least cost route in the case of all the *LAMOR-PMA* protocols, these three protocols have about the same ratio of the number of RREQs forwarded to the number of RREQs initiated as *HEAP*. This is because the *LAMOR* protocols have various measures in place to control RREQ flooding. This means that *LAMOR-PMA* protocols can obtain more information to improve their performances without having to flood as many RREQs as required by *PANDA*. The average number of RREQ and RERR packets generated per connection for *PANDA* and *HEAP* are respectively 80% and 115% higher than the number for the *LAMOR-PMA* protocols. This is because *PANDA* and *HEAP* generate a higher number of route error packets, and *HEAP* generates more RREQs because a source node here may need to transmit a number of RREQs at an increasing power level until a forwarding node is found.
5.6 Conclusion

It is generally recognized that power-awareness and mobility-awareness are two important issues that need to be taken into account for routing in a wireless ad hoc network. To address these issues, three variations of a power aware multi-objective problem for route selection are first formulated for a static network. These aim to select a route based on minimizing the total transmission power and maximizing the minimum node battery power. The three variations differ in that each addresses an additional concern individually, i.e. (i) minimizing the maximum energy consumption in the past interval, (ii) minimizing the maximum ratio of forwarding energy used to total energy used thus far, and (iii) optimizing with the transmission requirements of each connection known. As using a standard optimization tool to solve these problems online in a real system will not be feasible, (especially in a large and/or mobile networks) three corresponding on-demand location aided multi-objective heuristic routing protocols
Chapter 5 Location-aided Multi-Objective Routing

(Power Aware) LAMOR-PAL, LAMOR-PAF and LAMOR-PA/TR are proposed. It is assumed that each node knows its current position so that this location information can be used in all the LAMOR-PA protocols. All LAMOR-PA protocols are on-demand routing protocols that require route discovery to obtain routes before actual data transmission begins. As the information to be collected for route selection may be overwhelming, the flooding of route request packet is confined within a proposed forwarding area that includes the source and destination node, and re-broadcasting is allowed only if the forwarding node is nearer to the destination. A forwarding node will forward the route request packet if the route carried is of a lower cost (and, additionally, if the route also satisfies the battery power constraint in the case of LAMOR-PA/TR). Route cost here is mapped from the objective function. This additional checking done will control flooding and aids the destination in selecting the eventual route. Performance studies conducted in static networks shows that all three heuristic LAMOR-PA protocols closely approximate their corresponding optimized schemes. All the LAMOR-PA protocols are found to perform better than HEAP and PANDA in terms of network operational time and total number of data packets carried within the network operational time.

Three variations of a power and mobility aware multi-objective problem for route selection are next formulated. These variations also aim to select a route based on minimizing the total transmission power and maximizing the minimum node battery power. They differ in that the first considers maximizing route lifetime, the second will not do so but will need route lifetime information for routing, and the third will optimize based on a system where the transmission requirement for a connection is known. Three corresponding heuristic on-demand location-aided routing protocols (Power and
Mobility Aware) LAMOR-PMAMO, LAMOR-PMALF and LAMOR-PMA/TR are then proposed. All LAMOR-PMA protocols will obtain route lifetime information during the route discovery process so that nodes can pre-empt link breakages. With only the location given, a motion prediction module is proposed to compute velocity and heading direction. Both parameters are required to compute link and route lifetime. The route cost used here corresponds to the corresponding objective function. In the case of LAMOR-PMA/TR, battery power and route lifetime constraints will be considered when selecting routes. Performance evaluation in a mobile network shows all LAMOR-PMA protocols perform very well in terms of network operational time, network lifetime, throughput and delay performance. They are shown to be energy efficient and are able to greatly reduce the amount of route error packets initiated when links break. The low ratio of the number of route request packets forwarded to the number of route request packets initiated show the effectiveness of the proposed flood control mechanism.
Chapter 6 Conclusion and Recommendations

In this chapter, conclusions regarding the research work reported in this thesis will be made in Section 6.1. Based on the conclusions made, some recommendations for further research will be suggested in Section 6.2.

6.1 Conclusion

Wireless communication and networking are so prevalent nowadays that people will expect to be able to communicate using their wireless devices as and when required. Wireless ad-hoc networking will enable them to do so even if no networking infrastructure is in place. This means that wireless nodes have to perform routing themselves. The three major concerns have been identified: limited node battery capacity, cooperation between nodes in carrying out networking activities, and routing based on satisfying multiple objectives that include power and mobility awareness objectives. In this thesis, on-demand routing protocols have been designed to address these concerns.

To make use of the limited battery power capacity efficiently, a *Power Aware AODV* (*PAW-AODV*) routing protocol has been proposed. Power aware features that have been integrated into the standard *AODV* routing protocol are: new node and route cost functions that are both dependent on the remaining battery power capacity of a node, allowing more high-powered routes (i.e. routes going through nodes with higher battery power level) to be discovered, and preventing routing from happening through nodes with very low remaining battery power capacity. The simulation results show that *PAW-*
Chapter 6 Conclusion and Recommendations

AODV can carry more packets (i.e. higher data throughputs) by virtue of its power aware mode of operation while providing substantially similar delay performance compared to standard AODV protocol, especially when a suitable hop count limit is imposed. Further mobility performance studies on both AODV and PAW-AODV protocols show that mobility can redistribute power usage, resulting in more packets being carried for both protocols. Further tests carried out to investigate the effect of setting different hop count limits show that setting an appropriate hop count limit for both AODV and PAW-AODV protocols helps conserve energy for carrying more packets at a later time. This is because going on shorter path will result in less nodes expending energy for forwarding data.

To enforce nodes to cooperate in parting with their limited battery resources for routing, a cost-credit system is implemented. A node will have to pay a cost to other nodes for forwarding its data packets. The credits earned by a forwarding node can be used to pay for the forwarding services provided by others, should this node need to communicate as a source node at a later time. A forwarding rule that allows a node to transmit as many data packets as possible using a limited power resource is formulated in a simple scenario. This forwarding rule is adapted in a more realistic scenario by integrating it in a proposed On-Demand Cost-credit routing (ODCCR) protocol for cooperative routing. ODCCR can be applied to power constrained ad hoc networks, which may be either static or mobile. ODCCR uses a route discovery process to obtain routes; the forwarding rule evaluated in a node will decide whether the node allows a route to be built via it. The route management module of ODCCR takes care of routes with forwarding nodes that have subsequently become unavailable due to deactivation (no more battery power left) or mobility. Performance comparisons in static scenarios show that, regardless of
Chapter 6 Conclusion and Recommendations

the node density, **ODCCR** gives a higher throughput due to a longer network lifetime, as compared to **AODV** and **RDOBR**. However, **ODCCR** will only outperform the other two protocols when an appropriate amount of initial credits is assigned. Since **ODCCR** uses a route management module that can effectively handle node mobility, simulation results show that **ODCCR** performs as well or better than **AODV** and **RBODR** in mobile scenarios.

The cost-credit system is further analyzed to formulate an optimization problem that aims to maximize the minimum battery power among all nodes when selecting routes for all the source nodes. Routes selected must enable source nodes to transmit as many data packets as possible. A centralized routing protocol called *max-min Power Credit Routing* or **mmPCR** is then proposed to handle this route selection task. Performance studies show that **mmPCR** performs better than **AODV** even though the cost-credit system is in place to constrain transmission. This is because **mmPCR** selects routes for all nodes based on maximizing the minimum battery power among all the nodes, thereby prolonging network lifetime and throughput. At the same time, **mmPCR** allows all source nodes to transmit as many data packets as possible. However, **mmPCR** does not perform as well as **ODCCR** (in terms of network lifetime and throughput within the network lifetime) because **mmPCR** is addressing the performance issues from the network point of view. **ODCCR** uses the forwarding rule to improve the individual node lifetime and throughput performance.

To address the two power aware objectives in static networks, i.e. maximizing the minimum node battery power and minimizing the total transmission power required to reach a destination, three variations of a power aware multi-objective routing problem
Chapter 6 Conclusion and Recommendations

are formulated. These can be solved using a known optimization technique. As it is not practical to employ an optimization technique to solve and select routes in large networks, three on-demand Location-aided Multi-objective Routing (Power Aware version) protocols or LAMOR-PA based on the proposed heuristics have been designed. Each proposed protocol can achieve results close to those obtained via optimization, and all three protocols perform better than other power aware routing protocols like HEAP and PANDA in terms of network operational time and total number of data packets carried within the network operational time. With the presence of location information, a flood control mechanism to control the flooding of control traffic during route discovery is proposed.

To address the additional mobility aware objective, i.e. route lifetime, in mobile networks, three variations of a power and mobility aware multi-objective routing problem are formulated. In a real scenario, only location information can be obtained in each node using GPS. A mobility prediction module is thus proposed to compute the speed and heading direction of each node. All three parameters are required to compute the link and route lifetimes. Three LAMOR-PMA (Power and Mobility Aware version) protocols are proposed for mobile networks. Performance studies show that the LAMOR-PMA protocols perform very well in terms of network operational time, network lifetime, throughput and delay performance. They are shown to be energy efficient and are able to greatly reduce the amount of route error packets initiated when links break. The low ratio of the number of route request packets forwarded to the number of route request packets initiated show the effectiveness of the proposed flood control mechanism.
Chapter 6 Conclusion and Recommendations

6.2 Recommendations for Further Research

There are some recommendations on further research activities that can be carried out. Further investigating the number of cost zones, the associated node’s available power range and assigned node costs proposed for PAW-AODV in this thesis is one possible approach. These parameters and values can be varied and studied as a performance tuning work to be done. An enhancement is to implement a module in PAW-AODV to determine the 90-percentile hop count value in a practical scenario. One proposal is to have a node track the hop counts of its self transmissions, and obtain the hop counts of other data transmissions via overhearing. The node can then compute the 90-percentile hop count value within a period or window of time and apply this to the next window. Further experiments on using different hop count limit values, i.e. 95 or 99-percentile hop count values, can be carried out as well.

mmPCR may not be applicable in mobile or large network; the need for a central server to exist adds on to the complication. A distributed approach of mmPCR, with an appropriate forwarding rule integrated, may be an interesting work that could be further studied. To complement the strength of a good forwarding rule, another max-min power routing method similar to that used by mmPCR could be formulated. This proposed strategy may be implemented in cluster mode so that it can be used effectively in mobile and large network as well. In cluster mode, the cluster head or a designated node can assume the central server role as discussed in the case of mmPCR.

One possible enhancement for all LAMOR-PA and LAMOR-PMA protocols is to study and derive other configurations of forwarding zones, with the aim of maximizing the
number of low-cost routes to be found while minimizing control traffic flooding during route discovery. Another enhancement for LAMOR-PMA protocols will be to improve on the motion prediction module so that the link and route lifetime can be more accurately computed. To improve the accuracy, the average and variance information of a link lifetime may be considered when computing the route lifetime.

Route and link lifetime here are dependent on node mobility only. Link lifetime, or the period of time that two nodes will stay connected, is also dependent on other physical layer factors such as fading, interference and radio propagation loss. A possible alternative to obtain link lifetime is to measure and predict the link quality or connectivity. The proposed LAMOR-PMA protocols can then be evaluated based on this definition of link lifetime.

Some performance tuning issues need to be studied further. The weights associated with each objective and with the weighted average speed and pause time deserve a detailed study so as to improve the performance of the proposed protocols. This performance study may lead to fixed weighted values for each different scenario. In this case, studying and formulating adaptive weights to be associated with each objective could be a potential future research work to be done. The performance studies of LAMOR-PA protocols can be extended to include analytical work that provides analytical performance results pertaining to the power aware heuristic schemes.
Author’s Publications


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Qualnet 3.7 User Manual (downloaded html manual), Scalable Network Technologies, Inc.


1304.

mobility prediction,” Mobile Networks and Applications, Vol. 6, Issue 4, August

[90] C. Bettstetter and J. Eberspacher, “Hop distances in homogeneous ad hoc
57, Issue 4, pp. 2286–2290.
Consider an infinite area with nodes uniformly and randomly distributed. Each source node can make a connection to any destination node that is within a distance $3r$ from it, where $r$ is the radio range. The probability that a source node will make a connection and transmit to a destination within $[0, r]$ is $(\pi r^2/9\pi r^2)$ or $1/9$. Similarly, the probability of transmitting to a destination within $[r, 2r]$ and $[2r, 3r]$ are $1/3$ and $5/9$ respectively. As consistent with the observations made in [90], a connection made to a destination within $[0, r]$ requires only one hop as a direct link can be established. Let $P_h$ be the $h$-hop transmission probability. Thus, the one-hop transmission probability is $P_1=1/9$. Here, connections that require more than three hops or $h>3$ will be dropped. As such, a connection made to a destination within $[r, 2r]$ requires either two hops where one relay node is required, or three hops where an alternative path that involves two relay nodes can be found when no two-hop path can be found, or will be dropped due to hop count limit exceeded. Similarly, a connection made to a destination within $[2r, 3r]$ requires either three hops where two relay nodes are required, or will be dropped due to hop count limit exceeded.

The two-hop case can only happen when the destination is within the range $[r, 2r]$. The spatial distribution of the nodes here is given by a random point process on an infinitely large system plane. A homogeneous Poisson point process of density $\rho$ nodes per unit area is applied here. With the Homogeneous Poisson Assumption, an elemental area $\Delta A$ contains at most one node. The probability of this node’s existence is $\rho\Delta A$, where $\rho$ is the density of nodes in the plane. Using polar coordinates for convenience, and selecting
Appendix A Derivation of Initial Route Cost Value for ODCCCR

An arbitrary source node at the origin, the probability of finding a node at point \((x, \theta)\) will be \(\rho x(\Delta x)(\Delta \theta)\). Let \(n = \frac{A}{x \Delta \theta \Delta x}\), where \(A\) is the large area in consideration,

\[
P(\text{no node in an area } A) = \lim_{\Delta \theta, \Delta x \to 0} \left[ (1 - \rho x \Delta \theta \Delta x) \frac{A}{\Delta \theta \Delta x} \right] = \lim_{n \to \infty} \left[ 1 - \frac{\rho A}{n} \right]^n = e^{-\rho A} \quad (A.1)
\]

\[
P(\text{at least one node in area } A) = 1 - e^{-\rho A} \quad (A.2)
\]

Referring to Figure A.1, for a small area \(x \Delta \theta \Delta x\) located at \((x, \theta)\) for \(0 \leq \theta \leq 2\pi\) and \(r \leq x \leq 2r\) (within the range \([r, 2r]\) of the source node \(S\)), to be able to reach the destination located at the small area for a possible two-hop transmission to occur, there must be at least one intermediate node located within the overlapping area as shown in Figure A.1. Note that the approach here is similar to that used in [90], and so is the overlapping area in the two-hop transmission case, which is \(A_{\text{olap}}(x) = 2r^2 \arccos\left(\frac{x}{2r}\right) - \frac{1}{2} x \sqrt{4r^2 - x^2}\). \(A_{\text{olap}}(x)\) for a small area located at distance \(x\) from \(S\) is the same for any value of \(\theta\). The probability of finding at least one relay node in the overlapping area, given that the destination is located in the small area \(x \Delta \theta \Delta x\) at \((x, \theta)\) is \(1 - e^{-\rho A_{\text{olap}}(x)}\). Given that a source can select a destination that is within a distance \(3r\) from it, the probability that it will select a destination node within \([r, 2r]\) and able to find a relay node for a connection to be made is the probability of a two-hop transmission, which is given by

\[
P_2 = \frac{1}{9\pi r^2} \int_{\theta=0}^{2\pi} \int_{x=r}^{2r} x d\theta d(x(1 - e^{-\rho A_{\text{olap}}(x)})) = \frac{2}{9r^2} \int_{x=r}^{2r} x(1 - e^{-\rho A_{\text{olap}}(x)}) dx \quad (A.3)
\]

For a high node density, the probability of finding at least one relay node in the overlapping area approaches one and the two-hop transmission probability \(P_2\) will be given by \(\frac{2}{9r^2} \int_{x=r}^{2r} x(1) dx = 1/3\).
The three-hop case can take place when the destination is within the range \([r, 2r]\) or \([2r, 3r]\). To obtain the three-hop transmission probability, an enumeration method is required as various pairs of overlapping areas need to be considered, now that two relays are needed. Given that a source can select a destination that is within a distance \(3r\) from it, the probability that it will select a destination node within \([r, 2r]\) and not able to find one relay node but two relay nodes for a connection to be made is denoted as \(P_{3,2r}\). The probability that it will select a destination node within \([2r, 3r]\) and able to find two relay nodes for a connection to be made will be denoted as \(P_{3,3r}\). Thus, the total three-hop transmission probability is \(P_3 = (P_{3,2r} + P_{3,3r})\). For a high node density situation, \(P_{3,2r} = 0\) and \(P_{3,3r} = 5/9\). So the total three-hop transmission probability in this case is \(P_3 = 5/9\).

Therefore, considering a high node density situation where the various \(h\)-hop transmission probabilities are \(P_1 = 1/9\), \(P_2 = 1/3\) and \(P_3 = 5/9\) for \(h=1,2,3\), and given that the route cost for a \(h\)-hop transmission is \(hc\), \(c_{r,0} = c \sum_{h=1}^{3} hP_h = (P_1 + 2P_2 + 3P_3)c \approx 2.5c\).