An Enhanced Cross-Layer Routing Protocol for Wireless Mesh Networks Based on Received Signal Strength

Ebenezer Olukayode Amusa
Institute for Research in Applicable Computing
University of Bedfordshire

A thesis submitted to the University of Bedfordshire, in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

November 2010
"In theory, there is no difference between theory and practice;
In practice, there is."
- Chuck Reid
Abstract

The research work presents an enhanced cross-layer routing solution for Wireless Mesh Networks (WMN) based on Received Signal Strength. WMN is an emerging technology with varied applications due to inherent advantages ranging from self-organisation to auto-configuration. Routing in WMN is fundamentally achieved by hop counts which have been proven to be deficient in terms of network performance. The realistic need to enhance the link quality metric to improve network performance has been a growing concern in recent times.

The cross-Layer routing approach is one of the identified methods of improving routing process in Wireless technology. This work presents an RSSI-aware routing metric implemented on Optimized Link-State Routing (OLSR) for WMN. The embedded Received Signal Strength Information (RSSI) from the mesh nodes on the network is extracted, processed, transformed and incorporated into the routing process. This is to estimate efficiently the link quality for network path selections to improved network performance. The measured RSSI data is filtered by an Exponentially Weighted Moving Average (EWMA) filter. This novel routing metric method is called RSSI-aware ETT (rETT). The performance of rETT is then optimised and the results compared with the fundamental hop count metric and the link quality metric by Expected Transmission Counts (ETX).

The results reveal some characteristics of RSSI samples and link conditions through the analysis of the statistical data. The divergence or variability of the samples is a function of interference and multi-path effect on the link. The implementation results show that the routing metric with rETT is more intelligent at choosing better network paths for the packets than hop count and ETX estimations. rETT improvement on network throughput is more than double (120%) compared to hop counts and 21% improvement compared to ETX. Also, an improvement of 33% was achieved in network delay compared to hop counts and 28% better than ETX.

This work brings another perspective into link-quality metric solutions for WMN by using RSSI to drive the metric of the wireless routing protocol. It was carried out on test-beds and the results obtained are more realistic and practical. The proposed metric has shown improvement in performance over the classical hop counts metric and ETX link quality metric.
To the 'Ancient of Days', from whom every good and perfect gifts come from.
Acknowledgements

I would like to thank my Director of Studies, Dr Osei Adjei, as well as other members of my supervisory team (Professor Jie Zhang and Dr Ali Mansour), for their continuous encouragement, invaluable guidance and support in my research work. I also want to thank the Vice Chancellor, Professor Les Ebdon and Professor Andrew Slade (former Pro Vice Chancellor Research & Enterprise) for their discretion in allowing my payment plan when I ran into financial difficulties earlier on in the program.

I also thank Professor Antonio Capone and the MobiMESI / ANT-Lab team (Stefano Napoli, Stefano Paris, Fabio Turati & Alberto Pollastro) for their support in providing the infrastructure for setting up my initial test-bed while I was on a research visit to ANT-Lab in Milan. My sincere appreciation also goes to Andrés Alayon Glazunov and Ákos Ladányi for their support while taking measurements on my final test-bed at CWiND in Luton. I also want to thank Professor Jean-Frédéric Wagen (University of Applied Sciences of Western Switzerland, Fribourg) and Dr Dayou Li for reviewing my thesis.

For particular support at various stages of my research, I would like to thank Emanuel Thomas, Professor Angus Duncan & Maureen Daley of the Research Graduate School, Dr Enjie Liu, Dr Peter Norrington, Dr Susan Etok, Mr Malcolm Foster, Mr James Aduemuyiwa, Mr & Mrs Obidinma of Treviglio, Italy, and Rev Victor Okusanya. My sincere appreciation also goes to Mr Sunday Adeniyi who was always there to encourage and motivate me.
The support and encouragements from my family members cannot be over emphasized. I thank my dear wife, Olubunmi Amusa (Olólúfẹ), for her patience, encouragement and understanding. Also to my new baby boy, Theophilus OhunjiniOluwa whose birth energised me at the tail end of writing this thesis. To my mum, Comfort Amusa and my sister Esther Lijofi, I cannot fully expressed my appreciation for their unfailing supports. And to my dearly beloved father, Matthew Adesoye Amusa who passed away on the 3rd March 2010, I want to acknowledge his fatherly care, admonitions and prayer supports which kept me going through the night seasons. And to other friends and family members who have contributed to the success of this program but which space will not permit me to mention all their names, I say thank you to you all and God bless you.

This research work was partly supported by GAWIND project (Grant number: MTKD-CT-2006-042783). The financial support received towards this research is gratefully appreciated.
Declaration

I declare that this thesis is my own unaided work. It is being submitted for the degree of Doctor of Philosophy (PhD) at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other University.

Name: Ebenezer Olukayode Amusa
Date: 15 November 2010

Signature:
# Contents

Abstract i  
Acknowledgements iii  
Declaration v  
Contents vi  
List of Figures xii  
List of Tables xvi  
Acronyms xviii  

1 Introduction 1  
1.1 Background ................................. 1  
1.1.1 Brief History of Wireless Mesh Networks (WMN) 1  
1.1.2 Strength and Limitations of WMN .............. 2  
1.1.3 Applications of WMN .......................... 3  
1.1.4 Routing in WMN ............................. 4  
1.2 Motivation for the study .......................... 4  
1.3 Aims and objectives of the investigation ............. 6  
1.3.1 Aims ....................................... 6  
1.3.2 Objectives .................................. 6  
1.4 Scope and limitation of the work .................... 7  
1.4.1 Hardware .................................. 7  
1.4.2 Routing approach ............................. 7
CONTENTS

1.4.3 Software & drivers ........................................... 8
1.5 Organisation of the thesis ..................................... 8

2 Literature Review ................................................. 10
  2.1 Research concepts ............................................. 10
    2.1.1 Overview of IEEE 802.11 Standard ..................... 10
    2.1.2 Multi-Hop Wireless Networks .............................. 11
    2.1.3 Multi-Hop Wireless Networks Routing Protocols .......... 11
    2.1.4 Wireless Network Routing Design Goals .................. 13
    2.1.5 Relevance of shortest path routing in WMN ............... 14
    2.1.6 Optimized Link-State Routing (OLSR) ................. 15
    2.1.7 OLSR Multi-Point Relay (MPR) Flooding ............... 17
    2.1.8 OLSR Topology Discovery ............................... 18
    2.1.9 Link quality routing approach in OLSR ................. 19
    2.1.10 Cross-Layer designs in Wireless Networks ............. 19
    2.1.11 Cross-Layer Resource Allocation ....................... 20
    2.1.12 Cross-Layer Routing .................................. 21
  2.2 Received Signal Strength Information (RSSI) .............. 22
    2.2.1 RSSI analysis from power & signal strength point of view 22
    2.2.2 RSSI measurements ...................................... 24
    2.2.3 Application of RSSI metric ............................ 25
      2.2.3.1 Channel access & wireless roaming ............... 25
      2.2.3.2 Rate adaptation .................................. 26
      2.2.3.3 Transmission power control ....................... 26
      2.2.3.4 Preconditioning .................................. 26
      2.2.3.5 Localization ..................................... 27
      2.2.3.6 Electromagnetic field monitoring ............... 27
    2.2.4 RSSI variability ....................................... 27
  2.3 Link Quality routing solutions by different approaches .... 28
    2.3.1 Solutions by theoretical approaches .................... 28
    2.3.2 Solutions by simulation ................................ 30
    2.3.3 Solutions by implementation ......................... 31
  2.4 Related Link Quality Routing solutions ..................... 32
3 Design & Framework of RSSI-aware Metric

3.1 rETT Plug-in Algorithm

3.1.1 Retrieval of the RSSI information

3.1.2 Updating the measure RSSI values and the averaged values

3.1.3 Moving Average Filters

3.1.4 RSSI dBm Conversion

3.1.4.1 dBm conversion for Atheros

3.1.4.2 dBm conversion for Symbol

3.1.4.3 dBm conversion for Cisco

3.1.5 WMN chipset received sensitivity limits

3.2 rETT Metric Design

3.3 Dynamics of rETT metric

3.3.1 Case 1: When loss probability is variable & RSSI dBm is less than -68dBm

3.3.2 Case 2: When loss probability is variable & RSSI dBm is greater than -68dBm

3.3.3 Case 3: When loss probability is maximum & RSSI dBm is greater than -68dBm

3.4 OLSR Protocol Enhancement

3.4.1 Benefits of using plug-in approach in OLSR
## CONTENTS

3.4.2 RSSI-based metric on OLSR ........................................... 69  
3.4.3 Topology formation process by rETT metric ................. 72  
3.5 Summary ................................................................. 73  

### 4 Experimental Setup & Implementation Environment 75  
4.1 Test-bed components .................................................. 75  
4.1.1 Test-bed Hardware Components .................................. 76  
4.1.1.1 Linux Box .................................................. 76  
4.1.1.2 Wireless Cards .............................................. 76  
4.1.2 Test-bed Software Components .................................. 79  
4.1.2.1 OLSR Daemon - olsrd ................................ 79  
4.1.2.2 olsrd Components ......................................... 80  
4.1.2.3 Compilation, configurations and operations .......... 80  
4.1.3 rETT implementations ............................................ 81  
4.2 OLSR node configuration ............................................. 81  
4.3 olsrd debug levels ..................................................... 84  
4.4 olsrd debug output ..................................................... 84  
4.4.1 The links table .................................................... 85  
4.4.2 The neighbours table .......................................... 87  
4.4.3 The topology table ............................................. 88  
4.4.4 Dijkstra’s table ................................................ 89  
4.5 Experimentation measuring tools .................................. 90  
4.6 Implementation Issues ............................................... 91  
4.7 Experimental Environments ......................................... 91  
4.7.1 Overview of the experimental environment ................. 92  
4.7.2 Wireless mesh node locations .................................. 93  
4.7.3 Morning measurement environment ......................... 95  
4.7.4 Afternoon measurement environment ....................... 96  
4.7.5 Night measurement environment ......................... 96  
4.7.6 Advanced Network Technologies Lab (ANTLab) Test-bed 97  
4.8 Summary ................................................................. 98
CONTENTS

A WLAN Cards Specifications 171
  A.1 TP-Link TL-WN650G ........................................ 171
  A.2 DWL-G650 .................................................. 173
  A.3 Wistron Neweb CM9 Atheros ............................. 174

B Typical OLSR Configuration file 178

C More RSSI Experimental Results 185
  C.1 Comparison of statistical data for the different environmental
      measurements ................................................... 186
  C.2 Noise and transmit power effects on the measured RSSI . . . 192

D OLSR Components 197
  D.1 OLSR Routing Protocol Structure ........................ 197
  D.2 Information repositories ................................. 197
       D.2.1 Link set ............................................. 198
       D.2.2 Multiple Interface Association Information Base .... 198
       D.2.3 Neighbour Set ....................................... 198
       D.2.4 2-hop Neighbour Set ................................ 198
       D.2.5 MPR Set .............................................. 198
       D.2.6 MPR Selector Set ..................................... 198
       D.2.7 Topology Information Base ......................... 199
       D.2.8 Duplicate Set ....................................... 199
  D.3 Neighbour discovery ....................................... 199
  D.4 Control traffic ........................................... 200
       D.4.1 OLSR Packet Format ................................ 200
       D.4.2 OLSR message types ................................ 200
  D.5 Route calculation in OLSR ............................... 201
  D.6 Multi-point relaying ..................................... 202
  D.7 Link state flooding ..................................... 203
  D.8 Cross-Layer design approach in OLSR .................. 204
List of Figures

2.1 A typical Wireless Mesh Network showing the routing region . . 12
2.2 Classical OLSR Packet . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.3 Flooding approach in OLSR compared with classical approach . 17
2.4 Neighbour discovery sessions using HELLO messages . . . . . 18
2.5 Cross-Layer proposals . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
2.6 Block diagram showing RSSI stage in a Wireless receiver . . . 23
2.7 Delivery ratio calculation through Hello communication . . . . 33

3.1 RSSI Filter with different filter constants . . . . . . . . . . . . . . 57
3.2 The sensitivity limits of Atheros CM9 chipset . . . . . . . . . . . 61
3.3 OLSR protocol enhancement by the use of plug-in . . . . . . . . 69
3.4 RSSI aware OLSR Hello & TC messages . . . . . . . . . . . . . . 71
3.5 The sensitivity limits of Atheros CM9 chipset . . . . . . . . . . . 73

4.1 Dell laptop with a D-Link PCMCIA wireless card . . . . . . . . . 78
4.2 Single Radio with CM9 mini-card on embedded device . . . . . 78
4.3 TL-Link PCI wireless card on Linux box . . . . . . . . . . . . . . . 79
4.4 Pictures of D109 & D109a Park Square, Luton . . . . . . . . . . 93
4.5 Wireless mesh node locations . . . . . . . . . . . . . . . . . . . . . . 94
4.6 D109 & D109a in the morning time . . . . . . . . . . . . . . . . . . 95
4.7 D109 & D109a in the afternoon time . . . . . . . . . . . . . . . . . 96
4.8 D109 & D109a in the night time . . . . . . . . . . . . . . . . . . . . 97
4.9 Environment at ANTLab Milano for the experiments . . . . . . 97

5.1 RSSI measurement taken at morning time . . . . . . . . . . . . . . 101
5.2 Histogram of RSSI data at morning time .................. 101
5.3 Autocorrelation of RSSI data at morning time ............... 102
5.4 RSSI measurement data taken at afternoon time .............. 103
5.5 Histogram of RSSI data at afternoon time .................. 103
5.6 Auto-correlation of RSSI data at the afternoon time .......... 104
5.7 RSSI measurement data taken at night time .................. 105
5.8 Histogram of RSSI measurement at night time ............... 105
5.9 Auto-correlation of RSSI measurement at night time .......... 106
5.10 Range, standard deviation and mean values of RSSI measure-
    ment data at different times of the day ..................... 108
5.11 The comparison of measurement data showing the differences
    in signal levels at different times of the day ................. 109
5.12 Comparison of measurement data showing the auto-correlation
    functions of the data .................................. 110
5.13 The comparison of measurement data showing the PDF super-
    imposed with the theoretical Gaussian PDF .................. 112
5.14 Artificial interference measurement environment on node 2 ... 113
5.15 Graphs of measurement data of a node under artificial interfer-
    ence with different transmit power settings .................. 114
5.16 Histogram of measurement data of a node under artificial interfer-
    ence with different transmit power settings ................. 114
5.17 Graphs of measurement data of a node without artificial interfer-
    ence with different transmit power settings .................. 115
5.18 Histogram of measurement data of a node under artificial interfer-
    ence with different transmit power settings .................. 116
5.19 Mean, standard deviation and range values of RSSI measure-
    ment data at different transmit power levels under artificial interfer-
    ence ........................................ 117
5.20 Mean, standard deviation and range values of RSSI measure-
    ment data at different transmit power levels without artificial interfer-
    ence ........................................ 119
5.21 Range and standard deviation comparison for RSSI samples
    without and with artificial interference ....................... 120
6.1 RSSI variability ............................................... 123
6.2 Layout of experimental setup for filtering experiment - scenario 1125
6.3 RSSI Filter with alpha = 0.01 .............................. 126
6.4 RSSI Filter with alpha = 0.125 ............................. 126
6.5 RSSI Filter with alpha = 0.25 ............................. 127
6.6 RSSI Filter with alpha = 0.375 ............................. 128
6.7 Layout of experimental setup for filtering experiment scenario - 2129
6.8 RSSI Filter with different values of alpha - 3 .................. 130
6.9 Experimental Test-bed 1 (4 Wireless Mesh Nodes) ............ 133
6.10 Traceroute output between from Node 1 to Node 4 ............. 134
6.11 Traceroute output between from Node 1 to Node 2 ............. 134
6.12 Traceroute output between from Node 1 to Node 3 ............. 135
6.13 OLSR daemon debug output 1 ............................. 135
6.14 Experimental Test-bed 2 (4 Wireless Mesh Nodes) ............ 137
6.15 OLSR daemon debug output 2 ............................. 137
6.16 Experimental Test-bed 3 (5 Wireless Mesh Nodes) ............ 139
6.17 OLSR daemon debug output 3 ............................. 140
6.18 Experimental Test-bed 3 (5 Wireless Mesh Nodes) ............ 142
6.19 Routing performances on the network with different alpha values 143
6.20 Bar chart comparing mean throughputs for different metrics 147
6.21 Bar chart comparing the delay measurement between the metrics 149

C.1 The comparison of measurement data showing the differences in signal levels at different times of the day - 2 ................................. 186
C.2 The comparison of measurement data showing the PDF superimposed with the theoretical Gaussian PDF - 2 .................. 187
C.3 The comparison of measurement data showing the auto-correlation functions of the data - 2 ................................. 187
C.4 Mean, standard deviation and range values of RSSI measurement data at different times of the day - 2 ................................. 188
C.5 The comparison of measurement data showing the differences in signal levels at different times of the day - 3 ................................. 189
LIST OF FIGURES

C.6 The comparison of measurement data showing the PDF super-imposed with the theoretical Gaussian PDF - 3 189
C.7 The comparison of measurement data showing the auto-correlation functions of the data - 3 190
C.8 Mean, standard deviation and range values of RSSI measurement data at different times of the day - 3 191
C.9 Graphs of measurement data of a node under artificial interference with different transmit power settings - 2 192
C.10 Histogram & PDF of measurement data of a node under artificial interference with different transmit power settings - 2 193
C.11 Mean, standard deviation and range values of RSSI measurement data at different transmit power levels under artificial interference - 2 194
C.12 Graphs of measurement data of a node without artificial interference with different transmit power settings - 3 195
C.13 Histogram & PDF of measurement data of a node without artificial interference with different transmit power settings - 3 195
C.14 Mean, standard deviation and range values of RSSI measurement data at different transmit power levels without artificial interference - 3 196

D.1 Neighbour discovery sessions using HELLO messages 200
D.2 MPR selection process in OLSR 203
D.3 Overview of OLSR information repositories 204
D.4 Cross-Layer approach in OLSR, adapted from Tønnesen (2004) 205
List of Tables

2.1 Cross-Layer Resource Allocation ................................. 21
2.2 Comparison between routing characteristics and metrics ........ 47

3.1 A typical IOCTL call output .................................. 52
3.2 RSSI to dBm conversion scale for Symbol ....................... 59
3.3 RSSI to dBm conversion lookup table for Cisco ................. 60
3.4 Received Sensitivity Limits for Atheros AR5212 chipset .......... 62

4.1 OLSR config file parameters .................................. 82
4.2 A typical IOCTL call output .................................. 85
4.3 A typical link table ............................................. 86
4.4 A typical neighbours table ...................................... 87
4.5 A typical topology table ......................................... 88
4.6 A typical Dijkstra's table ....................................... 89
4.7 Table showing the environment classifications for the experiments 95

5.1 Statistical values of RSSI data at morning time .................. 102
5.2 Statistical values of RSSI data at afternoon time ............... 104
5.3 Statistical values of RSSI data at night time .................. 106
5.4 Table comparing the range, standard deviation and mean values of measured RSSI data ........................................... 108
5.5 Table comparing the mean, standard deviation and range values of different power levels under artificial interference ...... 116
5.6 Table comparing the mean, standard deviation and range values of different power levels without artificial interference ...... 118
6.1 Table showing throughput analysis on the network with different values of alpha ............................................. 144
6.2 Throughput comparison between the metrics ............................................. 146
6.3 Comparison of the delay for the different metrics ............................................. 148

C.1 Table comparing the mean, standard deviation and range values of measured RSSI data - 2 ............................................. 188
C.2 Table comparing the mean, standard deviation and range values of measured RSSI data - 3 ............................................. 190
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>ADC</td>
<td>Analogue-to-Digital Converter</td>
</tr>
<tr>
<td>ALM</td>
<td>Airtime Link Metric</td>
</tr>
<tr>
<td>ANTlab</td>
<td>Advanced Network Technologies Lab</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad-Hoc On-Demand Vector</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>BOR</td>
<td>Buffer Occupancy Ratio</td>
</tr>
<tr>
<td>CCT</td>
<td>Clear Channel Threshold</td>
</tr>
<tr>
<td>CSC</td>
<td>Channel Switching Cost</td>
</tr>
<tr>
<td>CWiND</td>
<td>Centre for Wireless Network Design</td>
</tr>
<tr>
<td>DREAM</td>
<td>Distance Routing Effect Algorithm for Mobility</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination Sequenced Distance Vector</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>EETT</td>
<td>Exclusive Expected Transmission Time</td>
</tr>
<tr>
<td>ETT</td>
<td>Expected Transmission Time</td>
</tr>
<tr>
<td>ETX</td>
<td>Expected Transmission Counts</td>
</tr>
<tr>
<td>Acronyms</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>EWMA</td>
<td>Exponentially Weighted Moving Average</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hop Spread Spectrum</td>
</tr>
<tr>
<td>HAL</td>
<td>Hardware Abstraction Layer</td>
</tr>
<tr>
<td>HWMP</td>
<td>Hybrid Wireless Mesh Protocol</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Number Authority</td>
</tr>
<tr>
<td>iAWARE</td>
<td>Interference Aware Routing Metric</td>
</tr>
<tr>
<td>IF</td>
<td>Intermediate Frequency</td>
</tr>
<tr>
<td>IOCTL</td>
<td>Input Output ConTroL</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>LPD</td>
<td>Link Packet Delay</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MadWifi</td>
<td>Multiband Atheros Driver for Wireless Fidelity</td>
</tr>
<tr>
<td>MANet</td>
<td>Mobile Ad-Hoc Network</td>
</tr>
<tr>
<td>mETX</td>
<td>Modified Expected Transmission Count</td>
</tr>
<tr>
<td>MIC</td>
<td>Metric of Interference and Channel-Switching</td>
</tr>
<tr>
<td>MID</td>
<td>Multiple Interface Declaration</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output</td>
</tr>
<tr>
<td>MPR</td>
<td>Multi-Point Relay</td>
</tr>
<tr>
<td>NBLC</td>
<td>Normalised Bottleneck Link Capacity</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimized Link-State Routing</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>OSI</td>
<td>Open System Interconnection</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PLR</td>
<td>Packet Loss Rate</td>
</tr>
<tr>
<td>RANN</td>
<td>Proactive Route Announcement</td>
</tr>
<tr>
<td>rETT</td>
<td>RSSI-aware ETT</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comment</td>
</tr>
<tr>
<td>RLC</td>
<td>Residual Link Capacity</td>
</tr>
<tr>
<td>RSSI-MA</td>
<td>RSSI Moving Average</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Information</td>
</tr>
<tr>
<td>RT</td>
<td>Roaming Threshold</td>
</tr>
<tr>
<td>RTT</td>
<td>Per-hop Round Trip Time</td>
</tr>
<tr>
<td>SDL</td>
<td>Simple Data Link</td>
</tr>
<tr>
<td>SFSR</td>
<td>Successful Frame Sending Rate</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal to Interference and Noise Ratio</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>TC</td>
<td>Topology Control</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>s-TDMA</td>
<td>Spatial-Time Division Multiple Access</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>USAP</td>
<td>Unified Slot Assignment Protocol</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WCETT</td>
<td>Weighted Cumulative Expected Transmission Time</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Networks</td>
</tr>
<tr>
<td>WMAN</td>
<td>Wireless Metropolitan Area Networks</td>
</tr>
<tr>
<td>WMN</td>
<td>Wireless Mesh Networks</td>
</tr>
<tr>
<td>WMR</td>
<td>Wireless Mesh Routers</td>
</tr>
<tr>
<td>WPAN</td>
<td>Wireless Personal Area Networks</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Wireless Mesh Networks (WMN) is an emerging and promising technology for the next generation of wireless networks. This architecture consists of Wireless Mesh Routers (WMR) connected in a multi-hop fashion to form a wireless backhaul which clients or end user terminals (laptops, mobile phones, blackberries etc.) can connect to. This provides access to the wireless backbone or interconnects with remote wired network for access to the Internet or other available services. WMN concepts are applicable to wireless access technology like IEEE 802.16 - Wireless Metropolitan Area Networks (WMAN) or WiMAX, 802.11 (Wireless Local Area Networks (WLAN)) or Wi-Fi and 802.15 (Wireless Personal Area Networks (WPAN)) or ZigBee.

1.1.1 Brief History of WMN

The first sector to fund research to investigate the application of mesh networking was the Military in the 1980s. This was to enable communications between war vehicles in and around the battlefield. But due to the high cost of hardware at that time, the interest became dormant until the 1990s when the cost of hardware became more affordable. However, the concept of ad
hoc networking has been in existence for over three decades and these include packet radio network (1972), survivable adaptive radio network (1980) and global mobile information system (early 1990s) (Bruno et al., 2005).

1.1.2 Strength and Limitations of WMN

The technical features of WMN are mainly attributed to the technologies deployed on the mesh routers and that of the access technologies. However, the strengths and limitations of WMN depend on the area of application. Recent researches in this area have highlighted some inherent advantages (Akyildiz and Wang, 2005; Bruno et al., 2005; Smith and Meyer, 2005) which are of great interest to the commercial and academic communities. These include:

- Lower deployment cost
- Better coverage
- Lower power consumption
- Easier administration.

Despite some of the advantages of using WMN technologies, there are equally some limitations which are open areas for further research. Some of the identified areas include:

- **Different packet delivery rates:** WMN uses a continuous range of radio link bit rates. This is determined by the Rate Control Algorithm implemented on the network and this process requires constant monitoring. Hence, this operation needs to be carefully determined for efficient use of the radio link which is not the case in traditional wired networks.

- **Lack of available spectrum:** The majority of mesh networks are operated using LAN technologies and they usually run on the unlicensed industrial, scientific and medical (ISM) band. There are lots of other consumer electronic devices like Bluetooth and other Wi-Fi devices sharing
1.1 Background

this spectrum. As a result, this spectrum is not adequate for the proposed range of applications. However, this limitation may be somewhat alleviated by the deregulation of the 5GHz band.

- **Interference problems:** Connectivity in WMN can fluctuate enough to interfere with data transfer, particularly in single-radio mesh nodes. But this issue can be mitigated by introducing channel diversity at the nodes. This can be achieved by putting two to three radios in each node.

1.1.3 Applications of WMN

WMN can be used to provide Internet connectivity to static and mobile users in a flexible and adaptive way and because the architecture can be auto-configurable and self-organising, the technology is becoming more popular and promising. Currently, there are commercial interests in adapting WMN in the area of home and community networking, high speed metropolitan area networking, enterprise networking and building automation (Akyildiz and Wang, 2005). The ranges of applications for WMN have motivated interests in optimising or re-inventing some of the existing protocol for WMN.

They can also be useful in areas where there are no pre-existing communication infrastructures like battlefields and disaster recovery. A recent use includes providing emergency services in times of disaster such as the January 19th 2010 earthquake in Haiti’s capital city, Port-au-Prince. A 7.0 magnitude earthquake stuck the city and severely damaged among other things, the telecommunication infrastructures like fixed landlines, cell phone networks and underwater high-capacity fibre optic cables. The Non-Governmental Organisations (NGO) and the rescue teams were able to rely on the long range Wi-Fi Network connected in mesh mode to coordinate rescue operation (IEEE Spectrum Podcast, 2010). They were able to save many lives in the process because these networks are relatively easy and quick to setup. Other advantages of WMN include flexibility, low cost of installation and maintenance, scalability both in density and
1.2 Motivation for the study

size. A survey by Akyildiz and Wang (2005) on WMN gives an overview of the technology and also highlighted various solutions and open issues.

1.1.4 Routing in WMN

Routing in WMN is performed solely within the WMR. This is very challenging due to the variability and unreliable characteristics of the wireless medium. Whilst in wired networks, the specifications of the transmission cables and few other parameters determine the state of the link. Minimum hop count or Shortest Path is the traditional routing approach used by WMN to determine the routes to take when transmitting packets from one end to another in a network. In the work by De Couto et al. (2003a), it was observed that when computing the routes, the chosen shortest path routes have less capacity links due to minimum hop counts rather than the best path based on link quality. Consequently, link capacity becomes congested because of packets retransmission as a result of a bad link, which will then generate more transmission errors. Hence it can be concluded that the shortest path is not sufficiently good enough to determine routing protocol.

With an efficient and intelligent routing protocol, congested routes or poor links can be avoided and nodes can choose the link with the best quality to transmit packets. This can result in achieving a faster, more reliable and higher throughput network. There is a need to develop and implement an efficient routing protocol that is aware of the state of the wireless medium. Any appreciable change in the state of the wireless medium should be able to trigger a change in the topology of a network in order to improve the network performance. This is the main focus of this research.

1.2 Motivation for the study

Cross-layer routing is an open research area. Classical routing protocols are primarily based on shortest path or hop counts and these are limited in perfor-
1.2 Motivation for the study

There is a need to increase the intelligence of routing protocols apart from the inherent layer 3 functionalities. The ability to incorporate other layers’ information or parameters with routing functionalities to improve performance is called Cross-layer routing.

Recently, researchers investigating this area have identified some solutions which use different approaches through simulation, theoretical and implementation methods. Some of the recent works using network simulation in this area confirmed that cross-layer designs can improve network performance (Rahman et al., 2009; Xiao-dong et al., 2009). Nevertheless, there are only a few researches in cross-layer routing with real implementations. Iannone et al. (2004) highlighted some of the inhibitions to cross-layer routing solutions through implementations and some of the major reasons include the following:

- Most of the current technologies are not able to support advanced features that can accommodate cross-layer design.
- Cross-Layer solutions are too complex to be implemented into the current operating system protocols.
- There are many impracticable assumptions in most of the recently proposed theoretical (Liu et al., 2007; Shen et al., 2007; Wu et al., 2009) and simulated solutions (Shang et al., 2009; Song et al., 2009; Xiao-dong et al., 2009) which are apparent in implementation solutions. For example, Network Simulator-2 (NS-2) which is a popular simulation tool among Wireless Networks researchers, has many deficiencies and imperfect channel modeling (Chen et al., 2007) which may have significant effects on the results obtained from such a simulator.

However, there are few works using the implementation approach; details of these will be discussed in the literature review. Most empirical methods introduce extra probes to achieve the cross-layer effect to adapt the routing process. For example, the work by Carrera et al. (2009) used three extra probes in estimating the expected transmission time for the WMN. Routing decision should be fast and light weight and this is one of the objectives of this work.
1.3 Aims and objectives of the investigation

This work investigates an efficient way of achieving a cross-layer routing solution based on the Received Signal Strength Information (RSSI) already embedded in 802.11 technologies. This is aimed at reducing the effect of extra probes on network performance. The routing protocol is extended to incorporate this RSSI-aware routing, and it is used to drive the metric of the routing protocol. The practicability of this investigation is carried out on implementation test-beds with wireless nodes.

1.3 Aims and objectives of the investigation

1.3.1 Aims

This work proposes and develops an enhanced cross-layer routing protocol extension for WMN which is intelligent and sensitive to the state of the wireless channel for efficient wireless routing, in order to improve network performance.

1.3.2 Objectives

- To study routing in WMN and analyse the shortcomings in the classical and current solutions to justify the need for a more efficient solution and approach.

- To propose an enhanced link quality routing metric based on cross-layer design that will involve the received signal strength information in the routing process for WMN.

- To study the characteristics of the RSSI samples in order to clarify and justify its suitability for the proposed approach.

- To incorporate the proposed solution into Optimized Link-State Routing (OLSR) to extend and enhance its routing process.
1.4 Scope and limitation of the work

- To implement the RSSI-aware metric on implementation test-beds in order to obtain realistic routing performance for the proposed solution.

- To evaluate and optimise the performances of the proposed RSSI-aware metric and compare with the classical and related link quality routing solutions.

1.4 Scope and limitation of the work

1.4.1 Hardware

The research focuses on efficient routing in WMN based on IEEE 802.11 technologies. WMNs based on WiMAX or ZigBee are not implemented on the test-beds. IEEE 802.11g WLAN cards are setup as the wireless nodes. Some of the analysis in the thesis include IEEE 802.11a and IEEE 802.11b but are not implemented on the mesh networks for the routing analysis.

WiMax and ZigBee cards are neither readily available nor cheap to purchase. Most of mesh network researches based on these technologies are often found within the industry for commercial purposes. However, IEEE 802.11g wireless cards are readily available in the market and are cheaper to acquire. The recent trend in the WMN researches in the academic community is to adopt implementation test-beds using off-the-shelf devices (Carleton University WMN, 2011; GATech WMN TestBed, 2011).

1.4.2 Routing approach

Proactive and reactive routing protocols in Wireless Mesh Networks are considered in the wireless routing analysis and literature review but the test-beds were setup for OLSR routing protocol. The results were compared with classical OLSR but not with Ad-Hoc On-Demand Vector (AODV) or Dynamic Source Routing (DSR).
Routing in WMN is most prominent at the mesh routers which are relatively stationary. This is similar to routing ad-hoc networks except that mesh nodes are typically stationary. Proactive routing protocols like AODV favours mobile nodes as routes are expected to be located quickly while for mesh nodes reactive approaches like OLSR are more beneficial. The concepts of proactive and reactive routing protocols are discussed further in the literature review.

1.4.3 Software & drivers

The RSSI obtained and analysed in this work is based on the Atheros chipset\(^1\). In particular, the AR5212 chipset is used for this investigation. Other chipsets by Symbol and Cisco are not implemented on the test-beds.

The implementation condition for the hardware equally holds for the software. Recent academic researches support the use of open source software and drivers (MIT RoofNet WMN, 2011; Riggio et al., 2008). Cisco and Symbol chipsets use proprietary drivers for implementation while Atheros chipsets work with open source drivers, e.g. Multiband Atheros Driver for Wireless Fidelity (MadWifi). This supports research activities and avoids the bottleneck imposed by using proprietary software.

1.5 Organisation of the thesis

Chapter Two commences with the background information on the subject area and explains briefly some of the concepts in the thesis. The literature of related works is also discussed. The literature review is arranged into two categories; firstly, the wireless routing solutions by approaches, i.e. theoretical, simulation and implementation. The second category discusses the related link-quality routing solutions with some of them adopting the cross-layer approach.

\(^1\)http://www.atheros.com/pt/wlan_core.htm
Chapter Three describes the methodology of this research which includes the design and the framework of the proposed metric. The details of metric design and algorithm is discussed and analysed. The dynamics and the limitations of the cross-layer metric cross-layer design approach in wireless network is discussed. The chapter also describes the OLSR routing protocol enhancement as a result of the metric running as a plug-in. This is to adapt the routing process to the link conditions based on the parameters of the measured RSSI.

Chapter Four covers the experimental setup and the implementation environments for the experiments and routing implementations. The components of the wireless nodes and the associated configurations and settings of the drivers are illustrated. The description of the configuration parameters and their effects on the routing process is highlighted. The details of the OLSR debug outputs are described in this chapter. The experimental environments for the various conditions and different scenarios is defined and described also.

The details of the experiments performed are discussed in Chapter Five. These include the measurement and statistical analysis of RSSI samples under different link conditions. The effects of noise and transmit power on the received signal strength are also discussed. The concept of probability density function used in analysing the measured data is also addressed.

Chapter Six covers the implementation of the proposed RSSI-aware metric into the routing process. The implementation of RSSI filters and RSSI-based metric on OLSR is carried out in this chapter. The performance and optimisation of the routing solution is also documented here. The comparison of the network performance of the proposed metric to other related metric is investigated in this chapter.

Chapter Seven concludes the thesis. The research contributions and transferability of the research result is discussed. The areas of the research that can be developed further are highlighted here. The concluding remarks and some recommendations are also discussed in this chapter.
Chapter 2

Literature Review

This chapter discusses the literature, reviewing work and concepts related to this research. The first part gives a background to the subject, more in-depth definitions of the concepts and how they relate to this research. The second part comprises of related work that includes the classical solutions and some of the recent solutions in link-quality routing for Wireless Mesh Networks.

2.1 Research concepts

This section briefly discusses some of the concepts and their relevance to this research.

2.1.1 Overview of IEEE 802.11 Standard

IEEE 802.11 WLAN standard was adopted in June 1997 and revised in 1999 (IEEE, 1999). This is a widely deployed technology and has many applications for different environments including home, enterprise and public access networking. The main characteristics of IEEE 802.11 standard are simplicity and robustness against failures due to the distributed approach of its MAC protocol.
Due to the frequency ranges specified for IEEE 802.11 standards, various sources of interference can arise. Metallic objects can reflect radio signals and create interference as well as causing dead spots. Other sources of interference include electronic devices emitting radio waves of the same or close frequency band, such as microwave ovens and Bluetooth devices.

2.1.2 Multi-Hop Wireless Networks

This can be described as a collection of wireless mobile or static hosts forming a network without any established infrastructure or centralized administration. Due to the limited radio range of each mobile station, other nodes might be needed to forward packets to the destination devices. For such multi-hop wireless networks a large number of routing protocols have been designed. These protocols use many different approaches - from simple modifications of Internet routing protocols to more complex multilevel hierarchical schemes.

Figure 2.1 shows a typical WMN consisting of an access network, mesh backbone and an Ad-Hoc extension. The mesh backbone consists of mesh routers which is where bulk of the network routing takes place. The size of the network is the function of the number of mesh routers in the network. Because of the distributed nature of the mesh backbone, the efficiency of the routing in the mesh backbone can affect the overall performance of the network. This segment of the network is the focus area that this research investigates.

2.1.3 Multi-Hop Wireless Networks Routing Protocols

Routing protocols are usually classified as proactive (table-driven) and reactive (on-demand), depending on the way they react to topology changes. Belding-Royer (2004) gave an overview on this topic. A host using a proactive protocol, broadcasts routing-related information to its neighbour depending on the definition of the protocol. This information might be broadcast at a certain interval. It can trigger other mobile hosts to recalculate their routing tables.
2.1 Research concepts

Figure 2.1: A typical Wireless Mesh Network showing the routing region

and further spread more routing-related information. Given this property, the amount of information disseminated each time is typically proportional to the size of the multi-hop network.

A different approach is taken by reactive routing protocols. Reactive routing protocols try to find a route to another node of the network if needed, hence on-demand. Protocol traffic depends on the number of active stations in a WMN. With respect to wireless networking environment a proactive protocol may incur costs when constructing routes even if mobile hosts do not have such a need, so wasting the limited bandwidth. Many researchers have proposed using reactive-style protocols, which allow only routes to be constructed on-demand, i.e. when they are needed. The characteristics of proactive and reactive protocols can be integrated in various ways to form hybrid routing protocols.

Examples of proactive routing protocols include the Destination Sequenced Distance Vector (DSDV) (Perkins and Bhagwat, 1994) and Optimized Link-State Routing (OLSR) (Clausen and Jacquet, 2003) while examples for reactive routing protocols are the Dynamic Source Routing (DSR) (Johnson et al.,
Both proactive and reactive routing approaches offer different preferences. Hybrid routing approaches try to combine these advantages in various ways. A proactive behaviour might be preferred given a certain set of circumstances, whilst, a reactive one might be preferred in another set of circumstances. That way the protocol can react flexibly to the characteristics of a given network. Zone Routing Protocol (ZRP) (Haas and Pearlman, 1998) and the Distance Routing Effect Algorithm for Mobility (DREAM) (Basagni et al., 1998) are examples of hybrid approaches.

### 2.1.4 Wireless Network Routing Design Goals

There are some design goals which need to be carefully considered when developing a routing protocol. It is also important to clarify the developmental goals for the routing protocol before choosing them. It is important to clarify the design choices of the protocols as these goals are derived from the characteristics of multi-hop wireless networks. A study by Belding-Royer (2004) analysing the extensive approaches to routing in Mobile Ad-Hoc Network (MANet) expressed the following design goals:

**Minimal control overheads:** Routing control information is requested or passed on by control messages. These control messages consume bandwidth and processing resources. These two resources are all limited in multi-hop wireless networks and should therefore not be used excessively. Only the minimum number of control messages that are needed to guarantee connectivity should be sent.

**Minimal processing overheads:** Computationally complex algorithms demand significant cycles in mobile devices. Thus, the more complex the algorithm, the more energy, hence battery power, is consumed.

**Multi-hop routing capability:** In a multi-hop wireless network, sources and destinations are not always within direct transmission range of each other.
2.1 Research concepts

due to the limited propagation range of the wireless medium. A multi-hop routing protocol must therefore be able to discover routes between source and destination.

**Dynamic topology maintenance:** Topology in a multi-hop wireless network is not always static. Routes or links in a multi-hop wireless network are subject to frequent changes. Therefore route changes must be handled quickly with a minimum of associated overheads.

**Loop prevention:** In all network types, loops are harmful for communication, as packet delivery is delayed or prevented. In multi-hop wireless networks this is even more important, because bandwidth is limited. Even a transitory routing loop will negatively influence the throughput of the network. This means that loops should be avoided at all times.

### 2.1.5 Relevance of shortest path routing in WMN

Shortest path routing is the basic approach to routing data packets in wireless networks. However, some recent research works have proposed modifications to this approach to improve the network performance (Fang et al., 2010; Yu et al., 2008). The underlying approach of shortest path routing is still the foundation of most routing protocols in wireless networks. OLSR and other MANet routing protocols run on Dijkstra’s algorithm. This is used to navigate packets between nodes in a wireless network.

Unlike in a wired network, the weight or cost of the link is a function of the link condition which is rather unstable. This in return makes the routing process more demanding as it requires constant updates on the link condition to resolve how the network topology is formed. Link conditions are generally measured by a metric. This is assigned to routes by routing protocols to provide measurable values that Dijkstra’s algorithm can use to select the path for packets on the network.
2.1 Research concepts

2.1.6 Optimized Link-State Routing (OLSR)

This routing protocol is documented in the experimental Request For Comment (RFC) 3626 (Clausen and Jacquet, 2003). For control traffic flooding it utilizes an optimization called Multi-Point Relay (MPR). All nodes collect topology information by receiving protocol control messages. The content of these messages will be saved in different information sets. The complete topology of the network can thus be reconstructed and a shortest path algorithm, like Dijkstra’s algorithm (Forouzan and Fegan, 2007), can be used to calculate the routing table. This routing table is then updated by an OLSR daemon according to the operating system of the node.

All OLSR traffic is sent in packets. Figure 2.2 shows the structure of a classical OLSR packet.

Some of the relevant fields in the structure are defined as follows:

**Packet Length (olsr_packlen):** The length in bytes of the entire packet which also includes the header.

**Packet Sequence Number (olsr_seqno):** This is incremented by one each time a new OLSR message is transmitted by this host.

**Message Type (olsr_msgtype):** This is an integer to identify the type of this message. Message type of 0 – 127 are reserved by OLSR while the 128 – 255 space is considered as “private”.

**Validity Time (olsr_vtime):** This indicates for how long after reception a node will consider the information contained in the message as valid.

**Message Size (olsr_msgsize):** The size of the message in bytes, including the header.

**Originator Address:** Main address of the originator of this message.

**Time To Live (ttl):** The maximum number of hops this message can be forwarded.
## 2.1 Research concepts

<table>
<thead>
<tr>
<th>1st Byte</th>
<th>2nd Byte</th>
<th>3rd Byte</th>
<th>4th Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>olsr_packlen</td>
<td>olsr_vtime</td>
<td>olsr_msgsize</td>
<td></td>
</tr>
<tr>
<td>olsr_msgtype</td>
<td>olsr_vtime</td>
<td>olsr_msgtype</td>
<td></td>
</tr>
<tr>
<td>ttl</td>
<td>hopcnt</td>
<td>seqno</td>
<td></td>
</tr>
<tr>
<td>originator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hTime</td>
<td>willingness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link_code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>message</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Header OLSR Message**

**HELLO message 1**

<table>
<thead>
<tr>
<th>Reserved</th>
<th>hTime</th>
<th>willingness</th>
</tr>
</thead>
<tbody>
<tr>
<td>hell_info[1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**HELLO info**

<table>
<thead>
<tr>
<th>link_code</th>
<th>reserved</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>neigh_addr[1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Topology Control packet 2**

<table>
<thead>
<tr>
<th>ansn</th>
<th>reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>neigh[1]</td>
<td></td>
</tr>
</tbody>
</table>

**neigh_info**

| addr | |
|------| |

**MID message 3**

| mad_addr[1] | |
|--------------| |

**HNA message 4**

| hna_net[1] | |
|-------------| |

**hnapair**

<table>
<thead>
<tr>
<th>addr</th>
<th>netmask</th>
</tr>
</thead>
</table>

---

**Figure 2.2:** Classical OLSR Packet
2.1 Research concepts

**Hop Count (hopcnt):** The number of times the message has been forwarded.

**Message Sequence Number (seqno):** This is incremented by one each time a new OLSR packet is transmitted by this host.

### 2.1.7 OLSR Multi-Point Relay (MPR) Flooding

Traffic forwarding is necessary to create multi-hop wireless networks and OLSR uses an optimization called MPR to achieve this process. This is used to decrease the number of retransmissions in a wireless network. To optimise the flooding process, every node selects a set of MPR and these MPR are selected in a way that all nodes of the network can be reached. The detailed description of this MPR selection algorithm is expressed in RFC 3626 (Clausen and Jacquet, 2003). Flooded packets are only retransmitted by the node’s associated MPR. This procedure ensures that packet retransmission is reduced and optimised.

![Flooding approach in OLSR compared with classical approach](image)

**(a) Classical**  |  **(b) OLSR**

**Figure 2.3:** Flooding approach in OLSR compared with classical approach
2.1 Research concepts

2.1.8 OLSR Topology Discovery

The primary part of topology discovery is neighbour detection; HELLO messages serve this purpose. Every node emits HELLO messages at regular intervals. First, empty HELLO messages are sent; that way a node announces itself to its neighbours. When receiving a HELLO message for the first time, the receiving node adds the sender to its neighbour set. If the sender already announces the receiving node, the sender has to be saved as a symmetric neighbour. If the sender has not yet announced the receiver as a neighbour, the sender has to be saved as an asymmetric neighbour. The new neighbour then has to be re-announced in the next HELLO message. HELLO messages do not have to be forwarded.

![Diagram of HELLO message exchanges](image)

**Figure 2.4:** Neighbour discovery sessions using HELLO messages

In order to get complete topological information, every node has to flood Topology Control (TC) messages. Every node has to announce its neighbours. TC messages have to be forwarded using the default OLSR forwarding algorithm. A node receiving a TC message has to save the contained information in its TC set. From the information in the TC message, the complete topology of the network can be reconstructed. This is why OLSR is a table-driven routing protocol.

As a wireless node can use more than one wireless interface with the OLSR protocol, those devices have to be announced as well. This information is
announced by Multiple Interface Declaration (MID) messages. These messages are only created by nodes that use more than one wireless interface for OLSR. They have to be forwarded by the default forwarding mechanism. A node receiving a MID message has to save the contained information in its interface association set.

### 2.1.9 Link quality routing approach in OLSR

These are routing approaches whereby the state of the communication channels are considered in forming network topologies and routing performance in WMN. More of the recent researches in this area are discussed latter in this chapter. Some are achieved through cross-layer techniques where some parameters of the wireless network layers other that classic routing process are used in adapting the routing process. Depending on the link quality design, these cross-layer resources can be from lower or upper layers of the network layer.

### 2.1.10 Cross-Layer designs in Wireless Networks

Cross-Layer design is a concept that allows information to be shared or transferred among the layers of the Open System Interconnection (OSI) model. Parameters from Physical and/or MAC layers of the wireless medium can be shared with higher layers like network or transport layer to aid in efficient allocation of network resources. Srivastava and Motani (2005) categorized the various cross-layer proposals into three major groups, and these are:

(i) Different communication between layers.
(ii) A shared database across the layers.
(iii) Completely new abstractions.

Figure 2.5 illustrates the various cross-layer proposals as highlighted above.
Layered architecture, over time, has proved to be an acceptable procedure for a baseline design. However, recently, new proposals have emerged to explore a much richer interaction between parameters across layers. Shakkottai et al. (2003) and Kawadia and Kumar (2005) while evaluating these proposals identified a trade-off between performance and architecture and it can be deduced that performance tends to be short-term focused while architecture tends to be long-termed focused.

When the structured layers are broken in a bid to optimize performance, precautionary measures need to be exercised to avoid complications due to unstructured approach to architecture development. It is possible to develop a system that is unstable and not scalable. This emphasizes the need to be cautious when using cross-layer designs. Srivastava and Motani (2005) identified the following open challenges for designers proposing cross-layer design ideas:

(i) The coexistence of different cross-layer design proposals.
(ii) The effect of cross-layer design on future innovations.
(iii) The significant impact of cross-layer design on network performance.

2.1.11 Cross-Layer Resource Allocation

Multi-hop wireless networks’ functionalities and resources are strictly interdependent due to the shared nature of the wireless communication channel.
2.1 Research concepts

Resources that are traditionally handled at different layers are strictly coupled and inherent in wireless network operations. For example, in analyzing the capacity of wireless link, the interference level at the receiver is an important factor and this is a function of other resources like power control, rate, routing etc. These are resources that are handled in a distributive manner at different protocol stack layers and by different network devices.

The main objective of this process is to optimize network performance, i.e. maximize throughput, minimize delay and energy consumption, maximise reliability etc. This is achievable through effective sharing of relevant information among layers and joint control of resources allocation decisions.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>QoS requirements, application controls</td>
</tr>
<tr>
<td>Transport</td>
<td>Data Rate, Rate control, Handoff loss &amp; delay</td>
</tr>
<tr>
<td>Network</td>
<td>Routing decisions, Path</td>
</tr>
<tr>
<td>Data-Link</td>
<td>Spectrum sensing &amp; sharing, adaptive scheduling, Link delay, Time slots</td>
</tr>
<tr>
<td>Physical</td>
<td>Power control, modulations, Coding rate, channel allocation (OFDM, CDMA)</td>
</tr>
</tbody>
</table>

2.1.12 Cross-Layer Routing

Cross-layer routing is the process whereby certain parameters from other layers can be cooperatively used with Shortest-Path algorithm to adapt the routing protocol to the inherent channel or wireless medium conditions. These parameters may include the Signal-to-Noise Ratio (SNR), Signal Strength, Power
2.2 Received Signal Strength Information (RSSI)

The RSSI is a measure of received power in a radio signal and it is a metric used in a generic radio receiver technology. This is a parameter that is now becoming popular in IEEE 802.11 wireless networking. Fundamentally, this measure is achieved in the Intermediate Frequency (IF) stage before the IF amplifier, this is done in the baseband signal chain, just before the baseband amplifier. The measured output is often at DC analogue level and can also be sampled by an internal Analogue-to-Digital Converter (ADC) and the output codes from this can either be through the internal processor bus, peripheral or made directly available. Figure 2.6 illustrates how RSSI is obtained in generic radio receiver technology.

2.2.1 RSSI analysis from power & signal strength point of view

The fact that signal strength does not fade in a linear manner makes power measurement more complex; actually power fades inversely as the square of the distance according to the inverse square law (Goldsmith, 2005). From equation 2.1, power is expressed in accordance with the inverse square law which fades inversely as the square of the distance from the transmitting end. Hence the received power can be expressed as;

\[ P_r(d) = C_f \left( \frac{1}{d^2} \right) P_t \]  

(2.1)
2.2 Received Signal Strength Information (RSSI)

![Block diagram showing RSSI stage in a Wireless receiver](image)

**Figure 2.6:** Block diagram showing RSSI stage in a Wireless receiver

where \( P_r \) is the power received

\[ P_t \] is the power transmitted

\( C_f \) is the constant of proportionality and depends on the transceiver. In this case, it is a wireless medium and assumed a free space propagation model.

\( d \) is the distance.

There are other expressions of received power based on different channel modelling. The common ones are the two-ray ground channel model and the log-distance channel model. Santi (2005) gave the following expressions for the other variants of equation 2.2 based on different channel modelling. These are expressed as;

\[ P_r(d) = C_t \frac{P_t}{d^4} \]  

(2.2)

This model incorporates reflections into the free space propagation model, i.e. the receiver receives direct communication ray and the reflected ray. It is assumed that the distance \( d \) is much greater that the height of the transceivers. \( C_t \) is the characteristic of the transceiver for the two-ray ground model. The other one is the log-distance channel model which was derived from empirical
2.2 Received Signal Strength Information (RSSI)

...and analytical methods. It defines received power as:

\[ P_r(d) = C_t \frac{P_t}{d^\tau} \]  \hspace{1cm} (2.3)

In this approach, \( \tau \) is defined as the distance-power gradient.

2.2.2 RSSI measurements

RSSI can be measured in milliWatts (mW), dB-milliWatts (dBm) or as a percentage measurement. The dBm is a logarithmic measurement and this can be directly converted to and from mW values, power measurement in dBm is more convenient because the dBm values change linearly. IEEE 802.11 2.4GHz Frequency Hop Spread Spectrum (FHSS) standard (IEEE, 1999) specifies the parameter to have a value from 0 through RSSI\text{Max} and the vendors are at liberty to determine the upper limit of the measure provided it is a 1 byte value (255). The Atheros chipset uses a maximum RSSI value of 60, Cisco 100 and Symbol uses 31 and they are left with the flexibility to determine the granularity, level of accuracy and the range of RSSI value.

For the Atheros chipset, the RSSI value at 0% is -95dBm and 100% is -35dBm. To get a percentage value of RSSI, 95 is added to the dBm value and hence RSSI at 0% becomes 0 and at 100% becomes 60. From experience on working with the test-beds, an RSSI value of 40 or more typifies a very strong signal and should support 54Mbps bit-rate, while a value of 20 typifies a decent signal to transmit effectively and a value of 10 or less represents a weak signal.

For the purpose of this research, digital RSSI values are considered and this is presently the choice for most integrated transceivers, e.g. Symbol, Atheros, Cisco chipsets and many more. Benetazzo et al. (2009) estimates the incoming power signal \( p(k) \) based on the above scenario at the instant \( k.T_s \) where \( T_s \) is the sampling time and \( k \) is an integer. The estimated received power measure expressed as:

\[ \hat{p}(k) = A \frac{1}{L} \sum_{l=0}^{L-1} x^2(k - l) \]  \hspace{1cm} (2.4)
Where $\hat{p}(k) = \text{the estimate of } p(k)$,

$x(k - l) = \text{the voltage sample at the instant } (k - l).T_s$

with $l \in \{0, 1, ..., L - 1\}$ and

$L = \text{the number of the last acquired samples.}$

$\hat{p}(k)$ is typically expressed as a logarithm value, in dBm units assuming a 50Ω input impedance.

It is important to note that the process of measuring RSSI in the circuit is not yet standardized. Hence, manufacturers of electronic components are free to assign the minimum and maximum values based on their design. But they are required to specify the sensitivity limits on the component data sheets. As a result of this, different chipset manufacturers have different RSSI Max, e.g. Atheros Communication assigned 60 as maximum RSSI value, Cisco assigned 100 as RSSI Max and Symbol assigned 31. Details of the conversion to dBm range are reported in subsection 3.1.4.

### 2.2.3 Application of RSSI metric

RSSI has varied applications in modern communication systems, some of which are:

#### 2.2.3.1 Channel access & wireless roaming

Rudimentarily, RSSI is used when an adapter wants to transmit a packet to another interface. The RSSI of the neighbour would be measured and if the RSSI value is below a threshold value then the device knows the channel is empty, this value is the Clear Channel Threshold (CCT) and a certain RSSI value is associated with this. Also, RSSI value is also used in roaming. When a device is about to decide whether to change to another associated AP, this is determined when the signal level received from the connected access point drops below a threshold value, this value is called the Roaming Threshold (RT).
2.2 Received Signal Strength Information (RSSI)

The roaming device can decide which Access Point (AP) to connect to by comparing the measure RSSI value from the available APs.

2.2.3.2 Rate adaptation

With particular attention to IEEE 802.11, RSSI values are used to estimate the data rate between two wireless nodes. This is estimated by considering the signal to noise ratio at the receiver input connector and also by determining the appropriate modulation scheme to be used. This is the case with the received sensitivity of IEEE 802.11 wireless cards which will be discussed further in the next chapter. The value of RSSI of the other node measured from the reference node is used in the computation of the rate of packet transmission between the two nodes.

2.2.3.3 Transmission power control

In most second and third generation mobile systems, the devices are powered by battery. Power control is a significant issue and still an open issue for research. Presently RSSI values at the receiver’s end are good parameters used to safe energy by optimising the transmission power. When the signal is good as interpreted from a high RSSI value, less power is required to operate the device unlike when the signal is low and much power is needed to perform the required operations.

2.2.3.4 Preconditioning

This is used in matching the ADC input range with the estimated incoming power from the wireless receiver antenna. From the architecture diagram, the analogue preconditioning stage uses the analogue RSSI values to make this adaptation. This is use to control the automatic gain level of the ADC depending on the input signal.
2.2 Received Signal Strength Information (RSSI)

2.2.3.5 Localization

In a sensor network, RSSI is an important parameter in solving localization problems (Anzai and Hara, 2009; Rencheng et al., 2008). This can be used to determine the location or position of an object or people in a specified area. The received RSSI are estimates of the received power levels and hence can be computed to determine the distance between node under observation and the transmitter.

2.2.3.6 Electromagnetic field monitoring

RSSI is also useful in the distributed measurement of the electromagnetic field intensity in areas such as university campuses, stadia, roads, industries etc. RSSI readings from deployed wireless nodes in the designated areas can be used in the computation of this parameter.

More accurate measures of the RSSI values are required for the monitoring and localisation application and hence the algorithm involved in the measuring of these values is more complex that that required for the rate adaptation and the power control cases.

2.2.4 RSSI variability

Variability of the measured RSSI is a critical issue in this research. The RSSI value oscillates around a mean value depending on the amount of signal strength received. This is studied in detail in chapter 6. The main reasons for this are the channel multi-path propagation and fading and shadowing of the Radio Frequency (RF) channel. Other factors that could contribute to the variability of the RSSI are thus:

- **Antenna orientation**: The radiation pattern of each antenna may not be uniform. Also, the measured RSSI for two communicating nodes with
specific distance will not be the same if one or both antenna are changed with all conditions remaining the same.

- **Transmitter variability**: The transmitters’ characteristics are different even with the same configuration. When a transmitter sends a packet at a certain level of power, the real power is a value slightly more or less and this affect the value received at the destination node.

- **Receiver variability**: Similar to the transmitter issue, the sensitivity of the receiver cannot be 100% accurate. Some receivers from the same model have different chipsets and hence the performances are slightly different from others even when the parameters are kept constant.

### 2.3 Link Quality routing solutions by different approaches

This section discusses recent solutions in the area of link quality routing by different approaches. These include solutions by theoretical, simulation and implementation methods.

#### 2.3.1 Solutions by theoretical approaches

Wu et al. (2009) proposed an application-centric routing framework for real time video transmission over multi-hop wireless networks. Expected video distortion was used as the metric of the routing. This is slightly different to the network-oriented approach of considering delay, throughput, or packet loss rate as the driving point of routing metrics. The authors considered packet loss probabilities and packet delays while estimating the expected end-to-end distortion. Wu et al. (2009) developed an efficient routing algorithm with the routing metric in terms of the expected video distortion using dynamic programming solution for the optimization problem. They also developed a quality-driven cross-layer optimization framework to enhance the flexibility
2.3 Link Quality routing solutions by different approaches

and robustness of routing by the joint optimization of routing path selection and video coding. The proposed routing algorithm was compared with existing link Packet Loss Rate (PLR) and link packet delay Link Packet Delay (LPD) metrics. Their results showed that their proposed application-centric quality-driven routing algorithm achieved better end-to-end video quality than the existing network-centric metric approaches.

The other theoretical solution was proposed by Liu et al. (2007). The work optimised power and bandwidth allocation at the respective nodes of the multi-hop/multipath routing in a Multiple Input Multiple Output (MIMO) based WMN. The wireless links were assumed to be operating in orthogonal channels. The authors approached the problem by developing a mathematical solution procedure of combining Lagrangian dual decomposition, cutting-plane and the gradient projection methods. In verifying the algorithm, the authors obtained some mathematical results through simulations using a 15-node MIMO mesh network. The scenario examined the network topology performance and convergence process and observed high efficiency with well decoupled structure. This attribute made the algorithm an effective method for optimizing performance of mesh networks based on MIMO technology.

Theoretical or mathematical based solutions are useful in a scenario where the wireless environment or channel cannot be simulated or practically implemented. Sometimes, this is a faster way of resolving engineering or computing problems. The main issue with this approach is that the actual channel or interference conditions cannot be accurately solved mathematically. These are complex situations which should be considered in the mathematical solutions. In solving the solutions mathematically, some constraints are applied to simplify the solutions or rather to make the problem solvable. In reality, these conditions are present in the actual wireless medium and should be considered in predicting network performance.
2.3 Link Quality routing solutions by different approaches

2.3.2 Solutions by simulation

Simulation approaches are faster ways of creating wireless network environments and scenarios. Simulations have been of tremendous assistant in solving wireless network problems or predicting network performance. Simulation tools like NS-2 (Information Sciences Institute, 2010) and QualNet (Scalable Network Technologies, 2010) are prime in wireless network simulations.

Song et al. (2009) published a cross-layer multipath routing work for WMN. The authors considered congestion issues by using data-link layer information to adapt the routing protocol originated from DSR. It was a two-dimensional congestion-aware metric approach: Buffer Occupancy Ratio (BOR) and Successful Frame Sending Rate (SFSR). The specified objective was to improve network throughput by finding multiple less congested routes. The work was simulated using NS-2 on DSR. The authors reported a significant throughput increase and also observed that the packet delivery was more reliable.

The work did not consider energy consumption and ignored hardware problems. All events of link breakage were considered as a result of congestion. The simulation experiment also did not consider Transmission Control Protocol (TCP) traffic and hence any packet delivery strategy due to TCP traffic was not incorporated into the scenario.

Another simulation approach with QualNet simulation was carried out by Shang et al. (2009). The work focused on the cross-layer approach of improving the network throughput using AODV routing protocol. These scenarios involved using Spatial-Time Division Multiple Access (s-TDMA) wireless network. The MAC Unified Slot Assignment Protocol (USAP) was modified and used to adapt routing process on AODV. The cross-layer techniques helped in reducing communication overheads and this resulted in achieving higher network throughput. The results from the QualNet simulator also showed lower number of route request messages sent by the nodes and lower routing overheads.
2.3 Link Quality routing solutions by different approaches

Wireless network design, planning or optimisations based solely on simulation tools are not conclusive as many channel conditions are not perfectly modelled in simulations tools. Few researchers have attempted combining simulation tools with other channel modelling applications to improve the accuracy of their investigations (Gutierrez and Cabrera, 2005; Qian et al., 2010). Chen et al. (2007) improved the accuracy of NS-2 for their work by adding additional parameters like cumulative Signal to Interference and Noise Ratio (SINR) frame body capture to the Physical layer of the simulator to improve the accuracy of the tool. A similar approach was introduced to QualNet network simulation in the work by Latkoski et al. (2010). The authors combined Simple Data Link (SDL) protocol developer to tune QualNet simulator to more realistic working conditions. The effect of this extension was evident in their results while simulating wireless heterogeneous IEEE 802.21-based network.

2.3.3 Solutions by implementation

Wireless network investigation by implementation or test-bed is by far the most realistic option. The limitations are obvious: they are not easy to set up and scenarios are not easily modified. Also, most advanced features needed for the investigations may not be supported or easily achieved on system architecture. It requires the operators to have a level of expertise with the experimental equipments and the operating system that the nodes will run on. Most test-beds are run on Linux or UNIX as they are open source and the software components are available to be modified to suit different purposes.

Carrera et al. (2009) focused on correlating wireless link cost to capacity. This was an implementation work aimed at showing the better network performance of their cross-layer approach of Expected Transmission Time (ETT) estimation to the network approach of ETT estimations. To achieve this, the authors used a probing system which involved three types of probes: broadcast packets, a pair of unicast packets, and another set of unicast packets. These were to estimate the number of retransmissions, to estimate the link transmission bit-rate and to collect the cross-layer information respectively. The authors
quantified the correlation for the metric capability of tracking link capacity change over time and the metric capability of correctly ranking links according to their quality.

Their cross-layer approach results showed supremacy over the network approach estimation of ETT. This was illustrated in correlation in tracking link capacity variation over time. They also obtained better results when auto-rate was enabled compared to when auto-rate was disabled. The control experiment was the packet pair technique ETT estimation by Draves et al. (2004b) which their results suggested leads to poor metric capacity correlation.

The work from Carrera et al. (2009) broadcasted three extra probes on the WMN to achieve their cross-layer approach of ETT estimation. Routing decision should be fast and light weight; these extra probes will definitely engage processing resources and consume bandwidth. There is a need to study a better approach to achieving this; an efficient approach is to make use of information already embedded on the system to achieve the link capacity estimation. Introducing extra probes to achieving this task is a disadvantage to the routing protocol and to the overall network performance.

2.4 Related Link Quality Routing solutions

This section further discusses routing solutions for WMN with close attention being given to link quality cost metrics. It highlights recent contributions and developments which differ from the classical approach of the minimum hop count metric, which is considered as not good enough (De Couto et al., 2003a). This section also discusses the merits and demerits of the metrics of the related solutions. Finally, it compares routing characteristics with various routing solutions.
2.4 Related Link Quality Routing solutions

2.4.1 Expected Transmission Counts (ETX)

Expected Transmission Counts (ETX) metric was first implemented by De Couto et al. (2003b) and the authors aimed to achieve high end-to-end throughput. The proposed ETX metric accounted for the link loss ratio, the existence of links with asymmetric loss ratio and the interference between successive hops of multi-hop paths. ETX was calculated by using the forward and reverse delivery ratio of the link. The measured probability that a packet arrives at the destination is called \( d_f \) while the measured probability that the Acknowledgement (ACK) packet is received is the reversed delivery ratio \( d_r \). This is illustrated in Figure 2.7.

\( \text{Figure 2.7: Delivery ratio calculation through Hello communication} \)

The ETX for a wireless link between two points is calculated as,

\[
ETX = \frac{1}{d_f \times d_r}
\]  

(2.5)

Therefore, ETX for a link between two wireless nodes is,

\[
ETX_l = \frac{1}{1 - e_f}
\]  

(2.6)

where \( e_f \) is the frame error rate for the link.

The \( ETX \) for the wireless path is the sum of the ETX for each link in the path. This is calculated as,

\[
ETX_{\text{path}} = \sum_{l=1}^{n} ETX_l
\]  

(2.7)

In their work, the ETX metrics performs better than the traditional minimum hop count on DSDV (proactive) and DSR (reactive) routing protocols in terms
of the achieved throughput of the wireless links. The ETX metric is also good for large networks.

ETX was a major breakaway attempt from the traditional hop count metric but also has some major drawbacks. There is the possibility of an additional collision due to additional packet probing introduced by ETX. This can also lead to losing critical routing information when the data rates are high and could lead to the routing protocol choosing a sub-optimal path (Ni, 2008). Another issue with this approach is that the probability of packet losses was measured using HELLO messages. The reality is that HELLO messages are broadcast / probe packets which are relatively small in size (≈64 – 134 bytes) compared to actual transmitted data and most links even in bad conditions will acknowledge these packets except if the link is down or extremely bad. ETX is also load sensitive and this will cause route oscillation which in itself causes the current path to appear worse (Zaidi et al., 2009)

2.4.2 Per-hop Round Trip Time (RTT)

The RTT metric was implemented in Adya et al. (2004). The authors estimated the channel quality of the link by investigating the RTT metric. Each node in the network measures the round-trip time and to every neighbour, i.e. Carrier Select (CS) and CS-ACK. The exponential weighted averages of the sampled RTT termed Smoothed RTT are measured. These values are then used by the routing algorithm to form the network topology and the target is to select a path with the smallest sum of RTTs of the links, to route data packets in the network. Therefore, the metric is expressed as;

\[
SRTT = \alpha \times RTT_{new} + (1 - \alpha) \times SRTT
\]  

(2.8)

where \( \alpha \) the weighting factor for the metric.

A comparative study by Draves et al. (2004a) shows that although RTT performs better than hop-count, the ETX metric performs better than RTT. An-
2.4 Related Link Quality Routing solutions

other drawback with RTT is the route oscillation issue due to load-sensitivity as experienced with the ETX metric (Zaidi et al., 2009).

2.4.3 Modified Expected Transmission Count (mETX)

Koksal and Balakrishnan (2006) having observed the drawbacks in ETX, proposed an enhancement to improve on the metric of this WMN routing protocol. Their solution, mETX aimed to account for the channel variability because ETX only considered the average channel behaviour (Koksal et al., 2006). The authors tackled the problem by defining mETX as thus:

\[
m_{ETX} = \exp \left( \mu_{\Sigma} + \frac{1}{2} \sigma_{\Sigma}^2 \right)
\]

where \( \mu_{\Sigma} \) represents the average error probability while \( \sigma_{\Sigma}^2 \) represents the variability of the error probability.

Basically, mETX increase with \( \mu_{\Sigma} \) which is a representation of the average level of the channel bit error probability over a period of time. The packet-to-packet variability is measured as \( \Sigma_k \) which is captured in the metric by the term \( \sigma_{\Sigma}^2 \). Therefore, mETX metric can be used to account for the channel variability at the packet time scale. Similarly to ETX, mETX for the path is the addition of the mETX for the various links connected in the topology.

\[
m_{ETX_{\text{path}}} = \sum_{l=1}^{n} m_{ETX_l}
\]

The main challenge with mETX apart from the drawback inherited from the ETX metric is how to realistically model the channel. Furthermore, the implementation of the variability of the transmission channel cannot be realistically quantified. Also, mETX does not take into account the probability that the number of transmission can exceed a certain threshold, it only concentrates on optimising the links layer throughput. This is not sufficient as the chosen path may involve links with high loss rates.
IEEE 802.11s standard (IEEE P802.11s/D1.08, 2008; Wang and Lim, 2008), which is one of the future technologies for WMN, proposes a default path selection protocol termed Hybrid Wireless Mesh Protocol (HWMP). This routing protocol uses MAC addresses to compute routes and to forward packets based on a radio-aware link metric called ALM. This metric takes into consideration the expected loss rate (as in ETX) and the quality of the link. The metric is defined as:

\[ C_a = \left[ O + \frac{B_t}{r} \right] \frac{1}{1 - e_f} \]  (2.11)

where:
- \( r \) = transmission bit rate
- \( O \) = channel access overhead
- \( e_f \) = frame error rate for a test frame of size \( B_t \).

ALM calculates the quality of the link by estimating the packet loss probability and the bit rate of the link. This metric is designed to operate in two modes, i.e. Radio-Metric Ad-hoc On-demand Distance Vector mode and the Proactive Route Announcement Proactive Route Announcement (RANN) Mechanism mode. ALM like ETX aimed at avoiding overloaded links which is not the case with a hop-count metric which basically sets up routes based on the number of hops.

ALM is an improvement to hop-count and ETX and it is a future default routing protocol for IEEE 802.11, but it already has some obvious drawbacks. This metric performed poorly under multi-radio multi-channel WMN (Ghan-nay et al., 2009). Though it is a simple design, it does not consider interference on the link and this will mitigate on the performance of the routing protocol. Also, this metric does not consider the contention impact due to traffic from neighbouring nodes. This contending traffic may increase the packet loss and
end delay due to collision. This may also lead to reduction in the available bandwidth on the link.

### 2.4.5 Expected Transmission Time (ETT)

ETT was implemented in the work by Draves et al. (2004b) and this work aimed to improve on ETX metric and to optimise the routing performance in multi-hop wireless environment. ETT metric considers the loss rate and also captures the impact of link bandwidth variability on the selected path. The ETT for a link between two nodes can be expressed as:

\[
ETT_i = ETX_i \times \frac{S}{B}
\]  

(2.12)

where \(S\) = size of the packets and \(B\) = capacity or bandwidth of the link.

With respect to a wireless network with \(n\) number of nodes, the ETT for the whole path can be expressed as:

\[
ETT = \sum_{i=1}^{n} ETT_i
\]

(2.13)

The main challenge with this metric is the estimation of the bandwidth between the links. The authors used a method termed packet pair techniques and this is estimated by broadcasting to each neighbour two back-to-back probes, one small one (137 bytes) followed by a large one (1137 bytes). The neighbour measures the inter-arrival time between the two packets and reports this back to the sender. The bandwidth is estimated by dividing the size of the larger probe by the smallest delay sample obtained.

ETT performed more efficiently with single radio WMN. Campista et al. (2008) show in their study that ETT metric performs better than ETX under various conditions. However, in order to improve the network throughput, it
is possible to configure each interface to a non-overlapping channel to reduce the interference between channels. This improvement introduces two issues, namely:

- Intra-flow interference: This is the interference between intermediate routers along the same path of flow. For example, a packet is sent from source node 1 to destination node 3 but goes via intermediate node 2. The link between nodes 1 and 2 will interfere with the link between nodes 2 to 3. Practically, the two adjacent links will not operate at the same instantaneous time.

- Inter-flow interference: Unlike the intra-flow interference, this is the interference between neighbouring nodes competing for the same channel. For example, if there are two flow paths in a wireless network and one of the intermediate routers of one path is adjacent to another router from the other flow path. The flow from path 1 will interfere with the flow from path 2 and vice-versa.

ETT did not consider these factors and may choose a path which uses only one channel. Hence the authors extended ETT to handle WMN with multi-radio functionalities.

### 2.4.6 Weighted Cummulated ETT (WCETT)

Weighted Cummulative Expected Transmission Time (WCETT) is proposed by Draves et al. (2004b) and it is designed to incorporate wireless nodes with multi-radio and multi-channel functionality. This is primarily to increase system throughput, reduce end-to-end delay and also to achieve better load balancing. WCETT for the path with multi-radio, multi-channel wireless network is expressed as:

\[
WCETT(p) = (1 - \alpha) \sum_{link \ l \in p} ETT_l + \alpha \max_{1 \leq j \leq k} X_j \tag{2.14}
\]
where $\alpha$ is a parameter configured subject to $0 \leq \alpha \leq 1$.

The problem of intra-flow interference was addressed by WCETT by reducing the number of links using the same channel within the path flow. However, the problem of inter-flow interference which is determined by the number of nodes sharing the channels could not be addressed.

### 2.4.7 Metric of Interference and Channel-Switching (MIC)

Yang et al. (2005a,b) aimed to resolve the inter-flow problem of WCETT by proposing the MIC metric. This is designed to capture more information on the effective link share. MIC averages the time to transmit on a particular link over the minimum time to transmit over all the existing links. Therefore, Channel Switching Cost (CSC) is added to the metric to account for the channel diversity. For a network with $N$ nodes and a path $p$. MIC is estimated thus:

$$MIC(p) = \frac{1}{N \times \text{min}(ETT)} \sum_{\text{link } l \in p} IRU_l + \sum_{\text{node } i \in p} CSC_i$$  \hspace{1cm} (2.15)

where $N$ is the total number of nodes in the network and $\text{min}(ETT)$ is the smallest ETT in the network. The other two components of the metric are defined thus:

$$IRU_l = N_l \times ETT_l$$  \hspace{1cm} (2.16)

$$CSC_i = \begin{cases} w_1 & \text{if } CH(prev(i)) \neq CH(i) \\ w_2 & \text{if } CH(prev(i)) = CH(i) \end{cases} \quad 0 \leq w_1 \leq w_2$$  \hspace{1cm} (2.17)

where $N_l$ is the set of nodes that interfere with the transmission on link $l$. $ETT_l$ is the expected transmission time on link $l$. $IRU_l$ is termed Interference-aware Resource Usage, while $CH(prev(i))$ represents the channel assigned for
2.4 Related Link Quality Routing solutions

node $I$’s transmission and $\text{prev}(i)$ represents the previous hop of node $I$ along the path $p$.

This metric addressed the problem of non-isotonicity in WCETT and also takes the inter-flow interference into account. However, there are other limitations of this metric which include the following: The CSC component of the metric only uses two consecutive links to capture intra-flow interference. MIC counts the amount of interferers on the link by considering the position of the interfering nodes irrespective of whether they are involved in any transmission simultaneously with that link. MIC also assigns the same value to the factored-in interference of a link caused by each interfering node in the neighbourhood and this is not considering the fact that traffic loads at each node may be different from one another.

In terms of implementation, MIC presents a major drawback. The overhead required to calculate the up-to-date information of the ETT for each links can hamper the performance of the network especially under heavy traffic loads.

2.4.8 Interference Aware Routing Metric (iAWARE)

Subramanian et al. (2006) proposed iAWARE routing metric for multi-radio WMN to factor in inter-flow and intra-flow interference. The authors attempt to resolve part of the limitation of MIC metric by considering a more accurate interference model. Unlike MIC, iAWARE factors in the amount of traffic generated by interfering nodes. This metric focused on considering link-quality variation by using Signal to Noise Ratio SNR and the SINR. These are continuously used to update the metric according to the variation of the interference by the neighbouring nodes. This metric calculates the average time the medium is busy as a result of transmission from each interfering nodes and hence, the higher the interference, the higher the iAWARE value. Therefore, the authors expressed iAWARE is as:

\[
iAWARE(p) = (1 - \alpha) \times \sum_{i=1}^{n} i\text{AWARE}_i + \alpha \times \max_{1 \leq i \leq k} X_j \tag{2.18}\n\]

40
2.4 Related Link Quality Routing solutions

Where $\alpha$ is a tunable parameter and $j, k$ is the number of available orthogonal channels. To find paths with less intra-flow interference and to exploit the channel diversity $X_j$ is defined as

$$X_j = \sum_{\text{conflicting links } i \text{ on channel } j} \text{iAWARE}_i, 1 \leq j \leq k, \quad (2.19)$$

The driving metric for a given link $l$ is represented by

$$\text{iAWARE}_j = \frac{ETT_j}{IR_j} \quad (2.20)$$

Where

$$IR_j = \min(IR_j(u), IR_j(v)) \quad (2.21)$$

$$IR_i(u) = \frac{SINR_i(u)}{SNR_i(u)} \quad (2.22)$$

$$SNR_i(u) = \frac{P_u(v)}{N} \quad (2.23)$$

And

$$SINR_i(u) = P_u(v)(N + \sum_{\omega \in \eta(u) - v} \tau(w)P_u(w)) \quad (2.24)$$

Where $\eta(u)$ represents the set of nodes from which node $u$ can sense a packet, $\tau(w)$ represents the normalised rate at which node $w$ generates traffic averaged over a period of time.

From the metric of iAWARE above, under no interference condition $IR_j$ equals 1 and hence iAWARE equals ETT. Therefore, the introduction of $IR_j$ and the inclusion of SINR into the metric of iAWARE is a major achievement. It helps in solving the inter-flow interference-aware problem in WMN routing. This is lacking in previous ETX-based solutions like WCETT and MIC.

Even so, iAWARE has its own inherent drawbacks. As it is still a variant of WCETT, the non-isotonic issue still exists. Furthermore, iAWARE did not consider the cost of channel-switching delay which is rather complicated to calculate. Also, due to the sensitivity of this metric to link traffic and the
2.4 Related Link Quality Routing solutions

presence of interfering traffic among neighbouring nodes, it can affect link stability. This may cause frequent or haphazard changes of established paths and disrupt normal network operation and hence reduce network performance.

2.4.9 Exclusive Expected Transmission Time (EETT)

Jiang et al. (2007) considered the channel distribution on long paths while proposing the EETT metric. This was aimed at maximizing the end-to-end throughput by selecting multi-channel routes with the least interference. This is an improvement on ETT metric by considering channel diversity distributed on large path. For a given link $l$, the authors expressed EETT metric thus:

$$EETT_l = \sum_{\text{Link } i \in IS(l)} ETT_i$$ (2.25)

Where $IS(l)$ is the interference set which incorporates the link itself. The metric for a given path $p$ is calculated as follows:

$$EETT(p) = \sum_{\text{Link } l \in p} EETT_l$$ (2.26)

$EETT_l$ denotes the busy degree of the channel used by link $l$. This means that if there are more neighbouring links operating in the same channel as the link, the link $l$ may be delayed for a longer time before transmitting on that channel. Therefore, a path with a larger EETT indicates a more severe interference and more time will be required to complete the transmission over all links involved on the specified path.

The main drawback of EETT metric is that it does not consider intra-flow interference which is very significant in making efficient routing decisions on WMN. Also, due to the fact that the interference range is greater than the transmission range, this metric can be improved by considering the intra-flow interference for transmission over large path.
2.4 Related Link Quality Routing solutions

2.4.10 Normalised Bottleneck Link Capacity (NBLC)

Load balancing was the focus of Liu and Liao (2008) when proposing the NBLC metric for WMN. The authors aimed to increase the system throughput by evenly distributing traffic load among nodes, as well as among channels. This has a specific advantage with networks where links are heavily loaded. NBLC can be represented as the residual capacity of the bottleneck link on a path normalized to the path length.

$$NBLC(p) = \min_{\text{link } i \in p} \left( \frac{RLC_i}{CEBT_{i,p}} \right)^\gamma L$$  \hspace{1cm} (2.27)

Where

$$RLC_i = T_m - \max_{j \in \{i, I_i\}} (\text{BusyPeriod}_j[n])$$  \hspace{1cm} (2.28)

Residual Link Capacity (RLC), is the percentage of free-to-use channel air time on an outgoing link. This is the difference between a measurement period ($T_m$) and the maximum of busy period of interfering neighbours at location $I_i$. The $\text{BusyPeriod}$ is defined as the time spent in carrier sensing, transmission and reception. Therefore the driving metric for NBLC is defined as:

$$CEBT_{i,p} = \sum_{i \in \{I_x \cap P\}} ETT_i$$  \hspace{1cm} (2.29)

$CEBT_{i,p}$ termed as the Cumulative Expected Busy Time of a link $i$ on a path $p$ is the sum of the ETT values for the path’s link transmitting on the same channel as $i$ and interfere with the link $i$. $\gamma$ is a tuneable design parameter which indicates the probability of a packet being dropped by an intermediate node in path $p$ of the length $L$.

NBLC routing algorithm is designed to choose the path with the largest NBLC. This indicates a less loaded, shorter, more channel-diverse path which was expected to have a more favourable link quality. This metric is more suitable under heavily loaded network links condition. This may be as a result of the high number of connections or the high data rate of a considerable number of
communications. The main drawback of NBLC is the inability to factor in the inter-flow interference into the routing algorithm and this makes the metric slightly unrealistic.

2.5 Routing Metrics & Characteristics

This section compares different routing characteristics with routing metrics. The routing characteristics are briefly described and the corresponding metrics using these features are highlighted.

Number of hops
This is the classical approach for routing process in wireless network and the only metric using this approach is minimum hop count. Other metrics under discussion in Table 2.2 are link-quality metrics which consider other characteristics.

Loss Ratio
This is primarily designed for ETX and other metrics which are variants of ETX also incorporated this characteristic in their metrics. RTT and of course the hop counts are not based on loss probability.

Data Rate
These are used by metrics which adopt data rate estimation (in bits per seconds) in designing the link quality. This is introduced in ETT and hence all metrics derived from ETT incorporate this characteristic.

Link Capacity
This is estimated in ETT as well. The bit-rate derived from the ETT estimation represents the link capacity in the design. Similarly all variants of ETT use this routing characteristic.

Channel Diversity
This characteristic is applicable to metrics with multi-radio capability and these are WCETT, MIC, iAWARE, ETT and NBLC.
Intra-flow Interference
This is the interference inherent on the network due to intermediate nodes along the same path. Hop count metric is free from this problem. Also, ETT variant metrics with channel diversity are able to solve this problem except EETT.

Inter-flow Interference
This is otherwise interference due to neighbouring node on the network. It is a problem for hop count but ETX and other metrics derived from ETX are free from Inter-flow Interference. However, ETT, WCETT and NBLC have this problem.

External Interference
This characteristic affects all metrics under observation on Table 2.2 except iAWARE which specifically targets solving this problem in the design of the metric.

Channel Distribution
These are only addressed by EETT and NBLC. Other multi-radio metrics like WCETT, MIC, and iAWARE still do not incorporate channel distribution in their solutions.

Channel Variation
This is an attempt by wireless routing metric to avoid congested channels on the network. This is only incorporated in the mETX and iAWARE metrics.

Locality
This is a feature of the iAWARE metric. It is aimed at optimising localisation in the metrics to improve network performance. This is a feature other metrics presently lack.

Load balancing
This is the ability of routing metrics to be sensitive to the size of data on the network. This is a feature most utilised by metrics that are able to run their metrics in a multi-radio environment. These include iAWARE, EETT and
2.5 Routing Metrics & Characteristics

NBLC, WCETT and MIC are equally designed for multi-radio environment but are incapable of balancing load traffic on the network.

Agility
This is a feature of a routing metric to respond quickly to changes in the position of wireless nodes. Hop count is excellent with this characteristic as it responds quickly to mobile nodes. This is why MANets adopt minimum hop counts as the classical routing metric.

Isotonicity
This is a routing feature whereby two similar links have equal tendency to be chosen for the transmission of wireless packets. It is generally required for routing metrics to be isotonic (Yang et al., 2005a). However, WCETT and iAWARE still have isotonic’s problems.

Stability
There is no more stable metric than hop counts. Other link quality metrics are rather less stable due to the introduction of additional feature for the routing process which take into consideration the state of the link. This is not primarily a feature of the routing process.

Table 2.2 illustrates the relationship between routing characteristics and the routing metrics reviewed in this study.
Table 2.2: Comparison between routing characteristics and metrics

<table>
<thead>
<tr>
<th>Routing Characteristics</th>
<th>Hop Count</th>
<th>ETX</th>
<th>RTT</th>
<th>mETX</th>
<th>ALM</th>
<th>ETT</th>
<th>WCETT</th>
<th>MIC</th>
<th>iAWARE</th>
<th>EETT</th>
<th>NBLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of hops</td>
<td>✓</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Loss Ratio</td>
<td>☒</td>
<td>✓</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Data Rate</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Link Capacity</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Link Quality</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Channel Diversity</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Intra-flow Interference</td>
<td>✓</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Inter-flow Interference</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>External Interference</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Channel Distribution</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Channel Variation</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Locality</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Agility</td>
<td>✓</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Isotonicity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stability</td>
<td>✓</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
</tbody>
</table>
2.6 Summary

Most link quality solutions on WMN are by cross-layer routing solutions. It has been proposed and suggested that this approach is better than network-only routing architecture. Some of the researches justified their results through theoretical and simulation approaches and very few results have been verified through implementation or test-beds. These solutions need a more practical approach since theoretical approaches have many constraints which could not be incorporated in the architecture. Likewise, simulation tools for simulated solutions at most times failed to give accurate and realistic results due to imperfect channel modelling. Therefore, it is necessary that these routing metrics be investigated on a test-bed to achieve a more realistic methodology and more accurate result.

Each routing algorithm and metric has both advantages and disadvantages and depending on the operating load and channel conditions, they may perform differently. In a bid to resolve some routing issues, some of the solutions introduced other problems and thus limit the area of application of such metrics.

Furthermore, most of the cross-layer routing works through implementation works focused at achieving quality-aware routing by broadcasting extra probe packets to estimate the channel quality in order to adapt the routing protocol. This practice, as expressed in the study by Belding-Royer (2004), increases the control overheads. Control messages consume bandwidth, energy and processing resources, these are not in abundant supply in multi-hop wireless network and therefore need to be efficiently managed. Hence, a good routing protocol should be able to optimize the minimum number of control messages needed to guarantee connectivity. The routing protocol should also utilize this information intelligently to achieve an efficient routing protocol with enhanced network performance. This is the approach of the routing protocol extension in this research.

This chapter has conducted a literature review of different approaches in cross-layer routing solution and has also evaluated different metrics proposed from
recent and related works. The next chapter is focused on the driving parameter for the proposed metric which is RSSI. Also, the relevant section of routing process that will provide the premise for RSSI implementation as a metric of the routing protocol is discussed.
Chapter 3

Design & Framework of RSSI-aware Metric

This chapter accounts for the methodology involved in this research. Firstly, it describes the steps involved in the proposed $rETT$ algorithm running as a plug on OLSR routing protocol. Also, the concept of the RSSI-aware metric based on transmission time estimation and its benefits over related approach is highlighted. Furthermore, the dynamics of the metric is discussed, particularly how it can adapt to different conditions dynamically depending on the state of the wireless channel. Lastly, the enhancement to the OLSR routing process is introduced. This includes the benefits of using the plug-in and the process of topology formation using this approach is described.

3.1 $rETT$ Plug-in Algorithm

In addition to the standard OLSR routing protocol running on the mesh nodes, an extension program running as a plug-in on the OLSR protocol was developed to carry out some initial investigation on the proposed algorithm. This program was later transformed and embedded into the OLSR routing protocol to develop the RSSI-aware OLSR. The program executed the following tasks,
which will be discussed in the following subsections:

1. Extracting the MAC Address and the RSSI of the neighbouring stations.
2. Updating the station lists and their RSSI values, then storing the old and the new values.
3. Filtering the RSSI value by an exponential moving average filter.
4. Converting the average RSSI value to dBm range.
5. Transforming the RSSI value to bit-rate based on the received sensitivity of the chipset.

3.1.1 Retrieval of the RSSI information

Algorithm 1 briefly describes the process involved in this operation. The station information which is also comprised of the RSSI values and the MAC addresses of the neighbouring nodes on the mesh network is extracted. This is achieved by Input Output ConTroL (IOCTL) calls by MadWifi driver for Atheros chipsets. IOCTL calls are Input/Output ConTroL calls to provide user-to-kernel interface to the hardware of the wireless cards. These are commands provided by the Network Interface Card (NIC)’s driver to communicate with the kernel of the NIC and hence return values that can be used for further processing. In this set-up, MadWifi 0.9.4 Linux driver was used for the wireless cards using Atheros chipset. Values like Media Access Control (MAC) addresses, rates, and RSSI information can be probed on the wireless cards. Different drivers have different syntax for probing this information depending on the product.

Table 3.1 shows some of the IOCTL calls by scanning the neighbouring wireless nodes and then returning the following information.

From the table, columns containing the ADDR and RSSI are retrieved directly from the NIC and these are raw MAC address and RSSI values. The
### 3.1 rETT Plug-in Algorithm

**Algorithm 1** - Extract RSSI from the mesh nodes

```python
neighbour_stations = empty list

THREAD update_average(t)

WHILE True

station_list = get_station_info_from_card

FOR s IN station_list

update_station(s.macAddress, s.RSSI)

ENDFOR

filter_old_stations

SLEEP t

ENDWHILE

ENDTHREAD
```

<table>
<thead>
<tr>
<th>MAC Addr</th>
<th>Time (s)</th>
<th>RSSI</th>
<th>RSSI-MA</th>
<th>RSSI-dBm</th>
<th>RSSI-Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:0f:3d:af:99:5b</td>
<td>17</td>
<td>47</td>
<td>46</td>
<td>-49</td>
<td>54</td>
</tr>
<tr>
<td>00:11:95:94:21:a2</td>
<td>17</td>
<td>44</td>
<td>44</td>
<td>-51</td>
<td>54</td>
</tr>
<tr>
<td>00:11:95:94:21:d0</td>
<td>17</td>
<td>30</td>
<td>30</td>
<td>-65</td>
<td>54</td>
</tr>
<tr>
<td>00:6b:4c:0d:0:a8</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>-76</td>
<td>48</td>
</tr>
<tr>
<td>00:6b:4c:0d:4:2</td>
<td>17</td>
<td>26</td>
<td>25</td>
<td>-70</td>
<td>54</td>
</tr>
<tr>
<td>00:6b:4c:0d:ab</td>
<td>17</td>
<td>19</td>
<td>19</td>
<td>-76</td>
<td>48</td>
</tr>
<tr>
<td>00:16:44:94:cc:aa</td>
<td>17</td>
<td>06</td>
<td>05</td>
<td>-90</td>
<td>09</td>
</tr>
<tr>
<td>00:19:d2:8cc:2:33</td>
<td>17</td>
<td>06</td>
<td>06</td>
<td>-89</td>
<td>09</td>
</tr>
<tr>
<td>00:18:de:25:ae:62</td>
<td>17</td>
<td>10</td>
<td>09</td>
<td>-86</td>
<td>18</td>
</tr>
<tr>
<td>00:21:5d:32:9c:92</td>
<td>17</td>
<td>11</td>
<td>11</td>
<td>-84</td>
<td>18</td>
</tr>
<tr>
<td>00:19:d2:51:9d:0f</td>
<td>17</td>
<td>08</td>
<td>08</td>
<td>-87</td>
<td>12</td>
</tr>
<tr>
<td>00:23:4d:ce:06:85</td>
<td>17</td>
<td>27</td>
<td>27</td>
<td>-68</td>
<td>54</td>
</tr>
<tr>
<td>00:12:0f:32:28:3b</td>
<td>17</td>
<td>04</td>
<td>04</td>
<td>-91</td>
<td>06</td>
</tr>
<tr>
<td>00:19:d2:63:b9:ed</td>
<td>17</td>
<td>06</td>
<td>06</td>
<td>-89</td>
<td>09</td>
</tr>
</tbody>
</table>

**Table 3.1:** A typical IOCTL call output
second column is the counter which are in seconds. Columns containing RSSI Moving Average (RSSI-MA), RSSI-dBm and RSSI-RATE are values returned from other functions of the plug-in program which include steps 3, 4 and 5 highlighted above. The details of these operations are discussed in subsequent subsections.

3.1.2 Updating the measure RSSI values and the averaged values

This operation is also included in algorithm 1. This performs two functions: first, to match the next RSSI value measured at the next cycle to the corresponding MAC address; second, to update the average RSSI value based on the filter parameter, which will be discussed later.

3.1.3 Moving Average Filters

The variability of RSSI is an important aspect of this research. This will be experimentally investigated in chapter 5 and implemented in the routing process in chapter 6. Meanwhile, it is necessary to investigate how to eliminate short-term fluctuations of the samples and extract the long-term behaviour of the samples to efficiently adapt the link condition for routing purpose. This can be practically achieved by passing the samples through a filter. The choice of the filter depends on the application and the objective of the filtering process.

Using an arithmetical mean is not appropriate on this occasion since this would require the summation of all of the samples divided by the number of samples. If the samples are 60 in number, collected every second then the mean is derived after 60 seconds. The arithmetic mean can be expressed thus:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

(3.1)
3.1 rETT Plug-in Algorithm

Thus a moving average is more appropriate, because the running average can be effectively calculated just after 2 seconds. Hunter (1986) proposed an Exponentially Weighted Moving Average (EWMA) which was expressed thus:

\[ S_t = Y(t-1)\alpha + (1 - \alpha)S(t-1) \] (3.2)

where \(0 < \alpha < 1\) and \(t \geq 3\). An alternative approach was proposed by Roberts (1959) where \(Y(t-1)\) was substituted for the current sample \(Y(t)\). This can be express as:

\[ S_t = \alpha \times Y_t + (1 - \alpha) \times S_{t-1} \] (3.3)

The alpha value \((\alpha)\), also called the smoothing factor or the filter constant, is critical to the implementation of the routing protocol and this is discussed later in the chapter. Meanwhile, a rough equivalent to the simple mean can be computed using the window length \((k)\). This is expressed as:

\[ \alpha = \frac{2}{(k + 1)} \] (3.4)

Hence the trend of the sample can always be followed and at some time be predicted. The value of the present average \((S_t)\) depends on the new value of the sample \((Y_t)\) and the last average of the sample \((S_{t-1})\). The average follows the trend of the new sample still but aware of the previous average.

Adapting Robert’s approach to filtering the RSSI samples, the EWMA for the RSSI samples can be filtered as:

\[ RSSI_{i(\text{avg})} = RSSI{i} \alpha + (1 - \alpha)RSSI_{i-1(\text{avg})} \] (3.5)
where $RSSI_{i(\text{avg})} = \text{present average value of the RSSI}$

$RSSI_i = \text{present sampled value of the RSSI}$

$RSSI_{i-1(\text{avg})} = \text{last average value of the RSSI}$

This is a tune-able filter with $\alpha$ as filter constant whose values range from 0 to 1. In a nutshell, when the $\alpha$ value is low, the output of the filter follows the value of the last averaged value. Conversely, a high $\alpha$ value gives an output that is sensitive to the present measured value of RSSI. The expression below describes the process. Algorithm 2 shows the implementation of the filter and also makes provision for new stations to be added to the network.

Algorithm 2 - Filter RSSI values and Update new nodes

FUNCTION update_station(macAddress, RSSI)

found = False

FOR s IN neighbour_stations

IF s.macAddress == macAddress

s.avg_RSSI = (1-alpha) * s.avg_RSSI + alpha * RSSI

s.avg_rate = RSSI_to_rate(s.avg_RSSI)

s.old = False

found = True

ENDIF

ENDFOR

IF NOT found

s = NEW Station

s.avg_RSSI = RSSI

s.avg_rate = RSSI_to_rate(s.avg_RSSI)

s.old = False

ADD s TO neighbour_stations

ENDIF

ENDFUNCTION
Figure 3.1 shows a typical output of the filtering operation on the RSSI samples and the effect of the filter constants on the measured sample. From the figure, it can be observed how the moving averaged value of RSSI responds to the trend of the presently measured value. When $\alpha$ is 0.01, the averaged value $RSSI_{i(\text{avg})}$ is less sensitive to the present value $RSSI_i$ but tends to follow the last averaged value $RSSI_{i-1(\text{avg})}$. When the alpha value is increased in figures 3.1(b) - 3.1(e), the sensitivity to the moving average is increased. By the time $\alpha$ value is increased to 0.5, the moving average almost follows the present RSSI value.
3.1 rETT Plug-in Algorithm

Figure 3.1: RSSI Filter with different filter constants
3.1.4 RSSI dBm Conversion

RSSI measurement is not yet standardised and there is flexibility in the way different NIC manufacturers represents this value. In order to use this metric for a scientific operation it is important to convert RSSI values to a range that can be referenced irrespective of manufacturer design. This research uses the Atheros chipset and this is implemented on the test-beds, but the conversion by Symbol and Cisco are also discussed in this subsection. The conversions from %RSSI values to dBm for Atheros, Symbol and Cisco are described in Bardwell (2004), as follows.

3.1.4.1 dBm conversion for Atheros

RSSI\_Max for Atheros is 60 (i.e.: x\% of RSSI\_Max=RSSI). Atheros RSSI output obtained through the IOCTL call is obtained as RSSI. With a slightly different approach to Symbol and Cisco, Atheros uses a formula to convert the RSSI value to dBm range. This is achieved by the algorithm using this equation.

\[
RSSI_{dBm} = RSSI - 95
\]  

(3.6)

It is necessary to also investigate how other common NIC manufacturers derive these values because it is possible in the future to change the hardware used on the implementation test bed. If this happens, then the function to convert to dBm range will change slightly to suit the design of the manufacturer.

3.1.4.2 dBm conversion for Symbol

Symbol uses 31 as RSSI\_Max. Therefore, to convert to dBm range, Table 3.2 is consulted to obtained the corresponding dBm value.
3.1 rETT Plug-in Algorithm

Table 3.2: RSSI to dBm conversion scale for Symbol

<table>
<thead>
<tr>
<th>RSSI</th>
<th>Conversion Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 4</td>
<td>-100 dBm</td>
</tr>
<tr>
<td>≤ 8</td>
<td>-90 dBm</td>
</tr>
<tr>
<td>≤ 14</td>
<td>-80 dBm</td>
</tr>
<tr>
<td>≤ 20</td>
<td>-70 dBm</td>
</tr>
<tr>
<td>≤ 26</td>
<td>-60 dBm</td>
</tr>
<tr>
<td>&gt; 26</td>
<td>-50 dBm</td>
</tr>
</tbody>
</table>

It should be noted that this gives a dBm range of -50dBm to -100dBm but only in 10dBm steps

3.1.4.3 dBm conversion for Cisco

Cisco conversion is more granular than the previous two conversions. RSSI Max for Cisco is 100. Therefore the RSSI ranges from 0 to 100 (101 steps) but the actual dBm range is from -113dBm to -10dBm with RSSI values from 94 to 100 having the same maximum dBm value of -10dBm. The lookup table for Cisco NIC is given in Table 3.3.
### 3.1 rETT Plug-in Algorithm

#### Table 3.3: RSSI to dBm conversion lookup table for Cisco

<table>
<thead>
<tr>
<th>RSSI</th>
<th>dBm</th>
<th>RSSI</th>
<th>dBm</th>
<th>RSSI</th>
<th>dBm</th>
<th>RSSI</th>
<th>dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-113</td>
<td>26</td>
<td>-86</td>
<td>3</td>
<td>-56</td>
<td>76</td>
<td>-32</td>
</tr>
<tr>
<td>1</td>
<td>-112</td>
<td>27</td>
<td>-85</td>
<td>2</td>
<td>-55</td>
<td>77</td>
<td>-30</td>
</tr>
<tr>
<td>2</td>
<td>-111</td>
<td>28</td>
<td>-84</td>
<td>51</td>
<td>-53</td>
<td>78</td>
<td>-29</td>
</tr>
<tr>
<td>3</td>
<td>-110</td>
<td>29</td>
<td>-83</td>
<td>52</td>
<td>-52</td>
<td>79</td>
<td>-28</td>
</tr>
<tr>
<td>4</td>
<td>-109</td>
<td>30</td>
<td>-82</td>
<td>53</td>
<td>-50</td>
<td>80</td>
<td>-27</td>
</tr>
<tr>
<td>5</td>
<td>-108</td>
<td>31</td>
<td>-81</td>
<td>54</td>
<td>-50</td>
<td>81</td>
<td>-25</td>
</tr>
<tr>
<td>6</td>
<td>-107</td>
<td>32</td>
<td>-80</td>
<td>55</td>
<td>-49</td>
<td>82</td>
<td>-24</td>
</tr>
<tr>
<td>7</td>
<td>-106</td>
<td>33</td>
<td>-79</td>
<td>56</td>
<td>-48</td>
<td>83</td>
<td>-23</td>
</tr>
<tr>
<td>8</td>
<td>-105</td>
<td>34</td>
<td>-78</td>
<td>57</td>
<td>-48</td>
<td>84</td>
<td>-22</td>
</tr>
<tr>
<td>9</td>
<td>-104</td>
<td>35</td>
<td>-77</td>
<td>58</td>
<td>-47</td>
<td>85</td>
<td>-20</td>
</tr>
<tr>
<td>10</td>
<td>-103</td>
<td>36</td>
<td>-76</td>
<td>60</td>
<td>-46</td>
<td>86</td>
<td>-19</td>
</tr>
<tr>
<td>11</td>
<td>-102</td>
<td>37</td>
<td>-75</td>
<td>61</td>
<td>-45</td>
<td>87</td>
<td>-18</td>
</tr>
<tr>
<td>12</td>
<td>-101</td>
<td>38</td>
<td>-74</td>
<td>62</td>
<td>-44</td>
<td>88</td>
<td>-17</td>
</tr>
<tr>
<td>13</td>
<td>-99</td>
<td>39</td>
<td>-73</td>
<td>63</td>
<td>-44</td>
<td>89</td>
<td>-16</td>
</tr>
<tr>
<td>14</td>
<td>-98</td>
<td>40</td>
<td>-72</td>
<td>64</td>
<td>-43</td>
<td>90</td>
<td>-15</td>
</tr>
<tr>
<td>15</td>
<td>-97</td>
<td>41</td>
<td>-71</td>
<td>65</td>
<td>-42</td>
<td>91</td>
<td>-14</td>
</tr>
<tr>
<td>16</td>
<td>-96</td>
<td>42</td>
<td>-70</td>
<td>66</td>
<td>-42</td>
<td>92</td>
<td>-13</td>
</tr>
<tr>
<td>17</td>
<td>-95</td>
<td>43</td>
<td>-69</td>
<td>67</td>
<td>-41</td>
<td>93</td>
<td>-12</td>
</tr>
<tr>
<td>18</td>
<td>-94</td>
<td>44</td>
<td>-68</td>
<td>68</td>
<td>-40</td>
<td>94</td>
<td>-10</td>
</tr>
<tr>
<td>19</td>
<td>-93</td>
<td>45</td>
<td>-67</td>
<td>69</td>
<td>-39</td>
<td>95</td>
<td>-10</td>
</tr>
<tr>
<td>20</td>
<td>-92</td>
<td>46</td>
<td>-66</td>
<td>70</td>
<td>-38</td>
<td>96</td>
<td>-10</td>
</tr>
<tr>
<td>21</td>
<td>-91</td>
<td>47</td>
<td>-65</td>
<td>71</td>
<td>-37</td>
<td>97</td>
<td>-10</td>
</tr>
<tr>
<td>22</td>
<td>-90</td>
<td>48</td>
<td>-64</td>
<td>72</td>
<td>-35</td>
<td>98</td>
<td>-10</td>
</tr>
<tr>
<td>23</td>
<td>-89</td>
<td>49</td>
<td>-63</td>
<td>73</td>
<td>-34</td>
<td>99</td>
<td>-10</td>
</tr>
<tr>
<td>24</td>
<td>-88</td>
<td>50</td>
<td>-62</td>
<td>74</td>
<td>-33</td>
<td>100</td>
<td>-10</td>
</tr>
<tr>
<td>25</td>
<td>-87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 3.1.5 WMN chipset received sensitivity limits

Every wireless NIC has received sensitivity limits based on the manufacturer’s design. This information expresses the achievable bit rates of the device with respect to the received signal strength of the neighbouring nodes. For Atheros chipset CM9 mini-PCI, the received sensitivities for the 802.11g are expressed in Figure 3.2. These values were also verified by the output of the sys-calls on the Linux embedded systems acting as WMN nodes in this research.
3.1 rETT Plug-in Algorithm

From the graph in Figure 3.2, it can be shown that a proportional increase in the received signal strength leads to a proportional increase in the achievable transmitting bit-rate for the wireless nodes.

The plug-in program implemented the $rssi_{dBm}$ transformation by using the process described in algorithm 3. At this stage, RSSI values for Atheros chipsets, which ranges from 0 - 60 (RSSI Max), have been filtered and converted to dBm range. The dBm values are then transformed to rates based on the sensitivity limits of the chipsets. The manufacturers of the chipsets on the wireless cards provides the sensitivity limits of each version of the chipset based on their design specifications. The received sensitivity for the NIC used in the implementation of the test-bed for this research are documented in Appendix A. Table 3.4 describes the sensitivity limits of Atheros chipset AR5212 which are used in the implementation test-beds for this study.

Lastly, algorithm 4 takes care of any nodes going dead during the operation to be safely removed from the network and from the routing plug-in operation.
Algorithm 3 - Convert and Transform averaged RSSI values to rates

FUNCTION RSSI\_to\_rate(RSSI)

\[
\text{RSSI}_{dBm} = \text{RSSI} - 95
\]

IF \( \text{rssi}_{dBm} \leq -88 \)
RETURN 6Mbps

IF \( \text{rssi}_{dBm} > -87 \) AND \( \text{rssi}_{dBm} \leq -85 \)
RETURN 9Mbps

IF \( \text{rssi}_{dBm} > -84 \) AND \( \text{rssi}_{dBm} \leq -83 \)
RETURN 12Mbps

IF \( \text{rssi}_{dBm} > -82 \) AND \( \text{rssi}_{dBm} \leq -80 \)
RETURN 18Mbps

IF \( \text{rssi}_{dBm} > -79 \) AND \( \text{rssi}_{dBm} \leq -76 \)
RETURN 24Mbps

IF \( \text{rssi}_{dBm} > -75 \) AND \( \text{rssi}_{dBm} \leq -72 \)
RETURN 36Mbps

IF \( \text{rssi}_{dBm} > -71 \) AND \( \text{rssi}_{dBm} \leq -69 \)
RETURN 48Mbps

IF \( \text{rssi}_{dBm} > -68 \)
RETURN 54Mbps

ENDFUNCTION

Table 3.4: Received Sensitivity Limits for Atheros AR5212 chipset

<table>
<thead>
<tr>
<th>From (dBm)</th>
<th>To (dBm)</th>
<th>Rates (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-95</td>
<td>-88</td>
<td>6</td>
</tr>
<tr>
<td>-87</td>
<td>-85</td>
<td>9</td>
</tr>
<tr>
<td>-84</td>
<td>-83</td>
<td>12</td>
</tr>
<tr>
<td>-82</td>
<td>-80</td>
<td>18</td>
</tr>
<tr>
<td>-79</td>
<td>-76</td>
<td>24</td>
</tr>
<tr>
<td>-75</td>
<td>-72</td>
<td>36</td>
</tr>
<tr>
<td>-71</td>
<td>-69</td>
<td>48</td>
</tr>
<tr>
<td>-68</td>
<td>-35</td>
<td>54</td>
</tr>
</tbody>
</table>
3.2 rETT Metric Design

This research work is a new cross-layer routing protocol metric using RSSI to adapt the process of OLSR routing protocol. The nodes employed on the test-beds are embedded GNU/Linux devices with wireless network cards with Atheros Communications chipsets.

The objective of the implementation experiments is to investigate how the received signal strength from the WMN can be used in the selection of link quality paths in multi-hop network and incorporate this into the framework of OLSR routing protocol to achieve an enhanced network performance under various links and channels conditions. This routing protocol extension should be sensitive to link quality changes and adapt the routing topology to accommodate any deficiency as a result of the changes.

The metric estimation for rETT is in the form of expected transmission time. The methodology of estimating the metric adopted the RSSI approach which is already embedded in the NIC for the wireless node. This is an efficient approach as it conserves the network resources by not broadcasting extra probes to estimate the link quality for the cross-layer routing.

It is imperative to note that control overheads and processing overheads are required to be kept at the barest minimum in order to maintain optimum net-
work performance. Control overheads as a result of extra probes from designs which are included in the control messages consume bandwidth. Also, computationally complex algorithms demand significant cycles in wireless devices. Thus, the more complex the algorithm, the more energy or battery power is consumed. The expected transmission time for the metric defined by Draves et al. (2004b) will incur an overhead which is a function of the number of nodes on the network. For example, in a network of $x$-nodes and each node with $y$ neighbours, the number of probes required for the transmission time estimation is a function of $xy$ (Esposito et al., 2008). Likewise, the extra probes required for the estimation of transmission time by $ScrRR$ metric by Aguayo et al. (2005) constitutes additional overhead which will compete with limited network resources and eventually hamper network performances.

Using the RSSI-aware ETT approach, the link metric equation can be expressed as:

$$r_{ETT} = ETX * \frac{S}{r}$$

where $S$ = size of the packet,

$r = \text{RSSI-to-rate value.}$

The $r$ value is based on the received sensitivity limits which are expression of the link capacity in this scenario.

The metric aims at the lowest Expected Transmission Time for any chosen link in the network and hence the RSSI-aware ETT (RSSI-aware ETT ($r_{ETT}$)) is thus:

$$r_{ETT_i} = \min(ETT_i)$$

where $r_{ETT_i}$ is the $r_{ETT}$ metric value of a link between two nodes.

Considering a wireless network with $n$ number of nodes, the RSSI-aware ETT for the whole path can be expressed as the sum of the ETT for each chosen links and this is expressed thus:

$$r_{ETT_p} = \sum_{i=1}^{n} r_{ETT_i}$$
3.3 Dynamics of rETT metric

This section discusses the dynamics of the rETT metric. This metric is driven by RSSI and the benefits of this metric are best appreciated when the link condition is poor. This is when the metric of the classic algorithm based on hop counts is inadequate. However, this metric is carefully designed and adaptive to function equally when the link quality is good or improves from a bad condition. This can be achieved without having to change the metric of the OLSR running on the WMN. These dynamics are illustrated in the following three distinct cases.

3.3.1 Case 1: When loss probability is variable & RSSI dBm is less than -68dBm

This is the default state of the metric and it estimates the quality of the link by the expression in equation 3.7. The transformed value of RSSI as a form of rate can increase or decrease the rETT value which the routing protocol uses in estimating how the topology is formed. The lower the rate, the higher the rETT value of the metric.

3.3.2 Case 2: When loss probability is variable & RSSI dBm is greater than -68dBm

In this condition, the RSSI values are generally above the threshold to effect any change in the transformed rate value. Hence, irrespective of the variable values of RSSI, the rate will return 54Mbps for the metric calculation. Here the metric assumes the signal strength of the link is at maximum. Therefore,
3.3 Dynamics of rETT metric

$S/B$ will remain constant and the only changes in the rETT metric will be as a result of the loss probability which is calculated through ETX estimation. rETT can then be expressed thus:

$$rETT_i = ETX_i \times k \quad (3.10)$$

where $k = (S/R)$ and $R$ is the maximum bit rate for IEEE 802.11g (which is 54Mbps).

### 3.3.3 Case 3: When loss probability is maximum & RSSI dBm is greater than -68dBm

This is an extreme condition wherein there is no packet loss. Every HELLO packet sent is received and acknowledged. When the loss probability is at maximum, the ETX value is 1.0, i.e. the probability of sending and acknowledging the test packets is 1. From equation 2.5, $ETX = \frac{1}{d_f \times d_r}$ and if there is no packet drop in either direction: then $ETX = 1$. Also, the signal strength is at the theoretically maximum value. Therefore, the rETT is constant for the link. Hence the routing process will adopt the classical mode for the metric calculation. This results in counting the number of hops from the source node to the destination and choosing the minimum hops for the packet transmission. This is the classical approach when link quality is not considered and OLSR RFC 3626 runs in this mode. Therefore, rETT will be expressed as:

$$rETT_i = k \quad (3.11)$$

where $k$ is a constant which is $S/R$ as described in subsection 3.3.2.

The three cases mentioned above can occur at any time in a WMN due to the transient nature of this technology. Therefore, it is important for the routing metric to be able to handle each of them adaptively as mentioned above. RSSI
filtering implementation in chapter 6 further illustrates how these conditions can be obtained in a typical WMNs.

3.4 OLSR Protocol Enhancement

Comprehensive documentation of OLSR protocol can be found in the RFC 3626 (Clausen and Jacquet, 2003). A few of the core and auxiliary functionalities relevant to this research work are documented in Appendix D. In order to achieve enhanced performance by extending the link quality feature of OLSR, this research focus on adopting a plug-in approach. This is achieved by running the rETT metric program to measure and process the RSSI data on the nodes in the WMN. The output of this operation is fed back into the main OLSR routing process to adapt the functionalities of the routing protocol.

The details of the plug-in program have been analyzed in the previous subsections. The metric used in forming the topology of the WMN nodes are solely determined by the functions in the plug-in. In this research the parameters are based on RSSI. The output of this program contributes to the overall performance of the routing protocol. The performance of the protocol can also be improved by fine tuning the operation of the plug-in instead of re-writing the whole components of the routing protocol. This is a faster and efficient approach to enhancing the performance of the routing protocol.

Routing application running on individual mesh nodes is a compiled version of the OLSR protocol and this is called olsrd. This term is used when the application running on the nodes is mentioned.

3.4.1 Benefits of using plug-in approach in OLSR

In designing and implementing olsrd, modularity is an important goal. This led to the idea of designing a plug-in interface to facilitate the link-quality
extension in this work. Because of this, it is necessary to highlight a few of the benefits of using this approach, such as:

- The plug-in program can be used in parallel by multiple processes in the main program as required by the design.

- It can provide new or additional functionalities to an existing application or process without the necessity to re-write the whole application or process.

- In some cases, plug-ins can be written in another language provided they can be compiled by the primary application.

- It is modular in operation, and in the case of OLSR when a newer version is released the amount of patching is reduced. It can easily be converted to a higher version of the application.

- There is no need to change most of the OLSR package in order to add a plug-in. Only the required linked files or dependent files need to be modified.

Figure 3.3 illustrates the connection between the plug-in program and the olsrd running on the a mesh node.
3.4.2 RSSI-based metric on OLSR

OLSR routing protocol is a network level (level 3) protocol which is driven by hop counts. To be able to achieve a link quality aware routing protocol, three things need to be carefully addressed:

- Generation of the OLSR packets
- Processing of the OLSR packets
- Data structure of the OLSR packets.

Figure 3.4 shows the structure of the RSSI-aware OLSR packets. This was incorporated into the routing protocol running on WMN to achieve the proposed link quality metric (rETT), in order to enhance the performance of the network. This requires modifying some OLSR files and linked files associated with the link quality aspect of the routing protocol. The plug-in (RSSI-rate.c) file is linked with the following files and the appropriate
3.4 OLSR Protocol Enhancement

header files: link_set, link_quality_packets, link_quality_route, neighbour_table, two_hop_neighbour_table, topology_control_set, olsr_cfg, olsr_types, packet, and process_package.

These listed olsr files are modified and declared to accommodate the plug in (extension file). All the classical olsr files with the extension are then compiled to generate the routing application (olsrd). The daemon and the configuration file are then run on every participating node on the network.
3.4 OLSR Protocol Enhancement

<table>
<thead>
<tr>
<th>1st Byte</th>
<th>2nd Byte</th>
<th>3rd Byte</th>
<th>4th Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>olsr_packlen</td>
<td>olsr_msgtype</td>
<td>olsr_vtime</td>
<td>olsr_msgsize</td>
</tr>
<tr>
<td>olsr_seqno</td>
<td>olsr_msg[1]</td>
<td>olsr_vtime</td>
<td>olsr_msgsize</td>
</tr>
</tbody>
</table>

**Header OLSR Message**

| olsr_msgtype | olsr_vtime | olsr_msgsize |
| origantor | ttl | hopcnt | seqno |
| message |

**rETT_Hello 251**

| Reserved | hTime | willingness |
| mac_addr | segue mac_addr | reserved | reserved |
| hell_info[1] |

**Hello info**

| link_code | link_type | neigh_type | link_quality | neigh_link_quality | neigh[1] | neigh_info |
| reserved | reserved | reserved | reserved |
| addr |

**rETT_TC_message 252**

| ansn | reserved |
| neigh[1] |

**neigh_info**

| link_quality | neigh_link_quality | rssi_to_rate |

**Figure 3.4:** RSSI aware OLSR Hello & TC messages

The rssi_to_rate information based on the proposed rETT metric is incorporated into the packet structure of OLSR to extend the routing protocol. This
3.4 OLSR Protocol Enhancement

makes the routing protocol aware of the RSSI functions. The routing metric is then calculated based on this information, basically, the routing metric of the OLSR is driven by this extension. The configuration files mentioned above were updated with the information accordingly and the enhanced routing protocol performances were analysed on the test-bed.

3.4.3 Topology formation process by rETT metric

Network topology is the logical interconnection of mesh nodes in a WMN. Sending network packet from a source to a destination may sometimes require the packets to go from node to node. Therefore, the routing decision to determine the appropriate links to form the end-to-end path is of importance to network performance. OLSR, as mentioned in chapter 2, is a proactive protocol which operates based on a routing table. The state of the link is estimated by the metric calculation based on the choice of metric by the protocol. The metric and other control information is relayed to other nodes on the network by the use of MPR flooding.

Figure 3.5 illustrates how the olsrd updates other mesh nodes on the network with the topology control information. This helps the routing algorithm to determine the topology of the network. The NIC of the mesh node obtains the RSSI information of the neighbouring nodes and passes this information to the plug-in program. The plug-in processes the information and returns the rate information (rssi_to_rate) to the olsrd. This information is used by the application to compute the routing table for the mesh node. Hence, the information is broadcasted to other nodes on the network for them to compute their routing table as well.
3.5 Summary

The process is carried out simultaneously by every node on the network and hence it is important for all the nodes to have similar configuration files for the routing process to converge. The details of the parameters and their effects are discussed in the next chapter.

3.5 Summary

This chapter documents the design and framework of the proposed RSSI-aware metric. The details of the $rETT$ plug-in algorithm are established. More importantly, the dynamics of the moving average filter on the RSSI samples are observed. Furthermore, the $rssi\_to\_rate$ transformation based on the sensitivity limit of Atheros chipset is incorporated into the design. The dynamics of $rETT$ and the adaptability to different channel conditions are also defined and proposed. This is important to this design as it makes the metric more robust.

The enhancement to the OLSR routing protocol which is a key objective in
this research was introduced. This involves the use of a plug-in and benefits of this approach are highlighted. The process of topology formation on the WMN based on the cross-layer routing approach is also highlighted. The next chapter introduces the implementation of the proposed metric by describing the experimental setup including the components and the environments where the experiments are conducted.
Chapter 4

Experimental Setup & Implementation Environment

This work focuses on implementing the WMN on test-beds rather than relying on simulation and analytic methods alone. Hence this chapter focuses on the system implementation part of this research. Hardware and software components used to achieve the implementation of the proposed methodology and routing solution are highlighted. Also the classification of implementation environments are defined. This is to ensure that the results obtained are practical and realistic. Some of the main components of the measuring tool are also discussed.

4.1 Test-bed components

The hardware and software components of the experimental setup are discussed in the following sections.
4.1 Test-bed components

4.1.1 Test-bed Hardware Components

The hardware components of this work are the Linux box and wireless cards which constituted the wireless node. Their specifications and basic configurations are discussed in the following sections.

4.1.1.1 Linux Box

The implementation test-bed is composed of different hardware components and off-the-shelf equipment with some modifications where necessary to facilitate this implementation. The first phase of the implementation was at Advanced Network Technologies Lab (ANTLab), with a test-bed with one laptop and three embedded systems. The laptop was a Dell Latitude D510 equipped with Intel Pentium M 1.73 GHz processor and 1 GB of RAM. The three embedded systems were VIA Nehemiah 1.0 GHz processors running on a 256 MB of RAM.

The laptop runs Ubuntu 8.04, which is a GNU free software operating software. The embedded devices run OpenWrt which is a GNU/Linux based firmware program for embedded devices such as routers and residential gateways. These are displayed in Figures 4.1 & 4.2.

The second phase of the research work was carried out at the Centre for Wireless Network Design (CWiND) laboratory at the University of Bedfordshire. The test-bed used comprised 5 RM desktop machines with 2.0 GHz Intel Processors and 1GB of RAM. The Linux boxes are all running on Ubuntu 0.84. This is shown in Figure 4.3.

4.1.1.2 Wireless Cards

The laptop is equipped with a D-Link DWL-G650 Cardbus (Atheros AR5212 chipset) wireless IEEE 802.11g interface while the embedded devices are equipped with Wistron Neweb CM9 Atheros 802.11a/b/g Dual-band mini-PCI 5004
4.1 Test-bed components

chipset interfaces. Since the wireless card features Atheros chipsets, the Mad-Wifi drivers 0.9.4 have been used. These drivers allow good management of the card, permitting operations such as power control via software and on-the-fly modification of wireless parameters.

The second test-bed used TP-Link TL-WN650 wireless cards. They ran on the same Atheros AR5212 chipset as the previous test-bed. The specifications of the wireless cards are found in Appendix A.

Wireless Cards Drivers and configurations

The wireless cards as describe above are using Atheros chipset and are using MadWIFI driver version 0.9.4 which can be downloaded from MadWIFI Project website\(^1\). This driver is stable and open source and its operation depends on the proprietary Hardware Abstraction Layer (HAL) which is available in binary form. This driver can be used to set up wireless node in station or adhoc mode. It can also be used to manipulate the power level, assign ESSID etc. to the wireless node. The main configuration command to setup the wireless node is `wlanconfig` which is a MadWIFI specific command; a typical command is:

```
wlanconfig athX create wlandev wifiX wlanmode adhoc
```

where X denotes the interface number for the node. Another important one is the `iwconfig` command which can be used to setup the wireless interface parameters like channel, power level, IP addresses etc. Other relevant MadWIFI commands can be found on both MadWIFI and Linux websites.

The following figures show the Dell laptop as wireless node and the GNU/Linux embedded devices with 5.8GHz Omni-directional antenna on single radio and the TP-Link card on Linux box.

\(^1\)http://madwifi-project.org/
4.1 Test-bed components

Figure 4.1: Dell laptop with a D-Link PCMCIA wireless card

Figure 4.2: Single Radio with CM9 mini-card on embedded device
4.1 Test-bed components

4.1.2 Test-bed Software Components

The system implementation of the routing protocol is discussed in the following subsections. Some of the network utility measuring tools are also discussed.

4.1.2.1 OLSR Daemon - olsrd

olsrd is an adhoc wireless mesh routing application. It is open source and can be downloaded freely from the OLSR website\(^1\). It is always released in two main formats, olsrd-x.x.x.tar.bz2 and olsrd-x.x.x.tar.gz, and recently there is an additional format which is olsrd-x.x.x.gpg. These are zipped files and can be unzipped for use on Linux box by using the following commands:

\[\text{tar jxvf olsrd-x-x-x.tar.bz2}\] (for the bz2 file extension)

and

\[\text{tar xvzf olsrd-x.x.x.tar.gz}\] (for the .gz file extension)

\(^{1}\text{http://www.olsr.org/}\)
4.1 Test-bed components

where $x.x.x$ denotes the file edition that was downloaded and to be unzipped.

4.1.2.2 olsrd Components

The source codes contain between 25 to 30 thousand lines and are readily available online [www.olsr.org]. The root folders are:

- **bin** – olsrd binary files are deposited here after linking
- **src** – It contains platform independent codes
- **src/Linux** – All Linux-only codes are deposited here
- **lib** – various plug-in codes are found here. They are sometimes linked with src and bin directories.
- **front-end** – This is for GUI front-end source. They are also linked with the src and bin directories.
- **files** – These are default configuration files and manuals.

**olsrd Versions** At the time the author started the implementation, the current olsrd version released was version 0.5.6 (October 2008). A number of issues arose with this version and because olsrd is open source, there are always some bugs with current editions as they have not yet been fully tested. Earlier versions such as 0.5.4 and 0.5.2 were also tested. In the end, version 0.4.9 was implemented because it was a bug-free, working version which is stable. So the rest of the experiment and the extensions are carried out running this version of olsrd on the Linux boxes.

4.1.2.3 Compilation, configurations and operations

After downloading, unzipping and compiling the olsrd codes, these codes can be set up on the Linux boxes and then the wireless interfaces used to run the daemon. For this to run properly, the same version of the daemon should be running on all the mesh nodes. Also, the same configuration files with the same configuration parameters must be running on all the mesh nodes on the network domain.
4.2 OLSR node configuration

4.1.3 rETT implementations

The plug-in file for the olsrd extension (RSSI-rate.c) was also compiled with the olsrd parent files and the various linkages and cross-linking as designed in the protocol extension. The filter constant was configured on /src/main.c and the value of this constant is embedded in this file before compiling the source code. Any time changes are made to this value all the olsrd files will need to be compiled again to effect the change on the olsrd operations. This operation is already described in the previous chapter under section 3.4 (OLSR Protocol Enhancement) and Figures 3.3 & 3.4 describe the framework and the various file linkages to accommodate the use of the plug-in as extension to the OLSR protocol.

4.2 OLSR node configuration

OLSR runs a daemon called olsrd which is basic application responsible for routing on the nodes on a WMN. RSSI-aware OLSR implementation runs in a likewise manner. To generate the olsrd for this modified version, the plug-in and the necessary linkages with the rest of olsrd files has to be complied and run on every mesh node on the network. It is required that the same compiled version of the olsrd and the configuration file (olsrd.conf) runs on every node as well. The configuration file can be found at /etc/olsrd.conf. There are usually 21 parameters and 18 sub options to be configured on this file. This may be increased if more than one interface is enabled on the node, i.e. if more than one radio is attached to the mesh node. Most of the parameters adopt the default value, i.e. they are left unchanged during the configuration process.

A typical olsrd.conf is found in appendix B. In a typical routing process, a few of the parameters need to be configured while others are left as default. For this research, some of the essential parameters of the configuration file are listed in Table 4.1. Also, the interface to be used for the routing should be
4.2 OLSR node configuration

specified. The routing application will not run if no interface is detected on the node.

Table 4.1 shows the configured parameters on the olsrd config files for the mesh nodes.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>UseHysteresis</td>
<td>no</td>
</tr>
<tr>
<td>LinkQualityLevel</td>
<td>2</td>
</tr>
<tr>
<td>LinkQualityWinSize</td>
<td>10</td>
</tr>
<tr>
<td>TCRedundancy</td>
<td>2</td>
</tr>
<tr>
<td>MprCoverage</td>
<td>3</td>
</tr>
<tr>
<td>Hello Interval</td>
<td>2.0 s</td>
</tr>
<tr>
<td>Hello Validity Time</td>
<td>8.0 s</td>
</tr>
<tr>
<td>TC Interval</td>
<td>2.0 s</td>
</tr>
<tr>
<td>TC Validity Time</td>
<td>15.0 s</td>
</tr>
</tbody>
</table>

It is important to briefly explain the functions of some of the relevant parameters in the configuration file and the influence they have on the routing process. Full details can be found on RFC 3626.

- **UseHysteresis**: Hysteresis adds more robustness to the link sensing but delays neighbour registration. This is usually set to “no”. Detailed information on this can be found on the RFC 3626. The option settings under hysteresis are only enabled if the parameter is set to “yes”.

- **LinkQualityLevel**: This determines the link quality scheme to be used.
  - 0 = do not use link quality, i.e. routing runs in “RFC3626 mode”
  - 1 = use link quality for MPR selection
  - 2 = use link quality for MPR selection and routing.

This option should be set to 1 or 2 if such a setting is used by all other nodes on the network domain.
4.2 OLSR node configuration

- **LinkQualityWinSize**: This specifies the window size to calculate the link quality. The default value is 10, meaning that *olsrd* looks at the last 10 packets to calculate the link quality. The number of packets lost can also be calculated from this value which is an arithmetic subtraction between the two values. Hence, link quality can be calculated.

- **TCRedundancy**: This value controls the TC redundancy used by the local node during the TC message generation.
  - 0 = The advertised link set of the node is limited to the MPR selectors.
  - 1 = The advertised link set of the node is the union of its MPR selector set and its MPR set.
  - 2 = The advertised link is the full symmetric neighbour set of the node.

The default value is 0 but for a link quality extension experiment like this, it is better to use the full symmetric neighbour set of the node.

- **MprCoverage**: This determines how many MPRs a node should attempt to select for every two-hop neighbour. The default value is 1, but 2 or 3 gives a better optimisation for the MPR scheme.

The following are configured under the interface section. They have to be configured for every interface enabled on the mesh node. The settings should be the same on all nodes on the network.

- **HelloInterval**: This sets the interval on which HELLO messages will be generated and transmitted on the specified interface.

- **HelloValidityTime**: This sets the validity time to be announced in HELLO messages generated by the node on the specified interface.

- **TCInterval & TCValidityTime**: Same as HELLO, this is the TC interval and validity time.
4.3 olsrd debug levels

Another feature of the configuration file that is very useful for the research is the debug output. DebugLevel has 10 different debugging levels, i.e. 0 to 9. This controls the amount of information olsr sends to the standard output (stdout) device, which in this case is the desktop. The default setting is zero which means the debugging features run in the background while level 9 returns the maximum debug output. For experimental and analysis reasons the debug level for this work is set to level 2. At level 2, most of the link quality features are reported in the output and these were used to analyse the performance in chapter seven.

There are two ways to set this parameter, primarily through olsrd.conf file or through the terminal by adding the debug option (olsrd -d 2) to the command line to invoke olsrd. It should be noted that the command line specifications override what is written in the olsrd.conf file; this is also applicable to all other parameters in the configuration file.

4.4 olsrd debug output

From olsrd version 0.4.8, which supports the ETX link quality metric, the level 2 debug output changes drastically. It is now easier to see what olsrd does and how it monitors the link quality to the neighbours on the WMN. This may consume a little of the CPU resources especially when olsrd is being run on an embedded system. More of these outputs are reported in chapter seven but their functionalities will be discussed briefly here. Table 4.2 shows a typical level 2 output from olsrd.
### 4.4 olsrd debug output

#### Table 4.2: A typical IOCTL call output

--- 19:27:45.51 -------------------------------------- DIJKSTRA'S

192.168.120.1:1.00 (one-hop)
192.168.120.3:1.00 (one-hop)

--- 19:27:45.51 ----------------------------------------- LINKS

<table>
<thead>
<tr>
<th>IP address</th>
<th>hyst</th>
<th>LQ</th>
<th>lost</th>
<th>total</th>
<th>NLQ</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.120.1</td>
<td>0.000</td>
<td>1.000</td>
<td>0</td>
<td>20</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>192.168.120.3</td>
<td>0.000</td>
<td>1.000</td>
<td>0</td>
<td>20</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

--- 19:27:45.51 ------------------------------------- NEIGHBOURS

<table>
<thead>
<tr>
<th>IP address</th>
<th>LQ</th>
<th>NLQ</th>
<th>SYM</th>
<th>MPR</th>
<th>MPRS</th>
<th>will</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.120.1</td>
<td>1.000</td>
<td>1.000</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>3</td>
</tr>
<tr>
<td>192.168.120.3</td>
<td>1.000</td>
<td>1.000</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>6</td>
</tr>
</tbody>
</table>

--- 19:27:45.51 -------------------------------------- TOPOLOGY

<table>
<thead>
<tr>
<th>Source IP addr</th>
<th>Dest IP addr</th>
<th>LQ</th>
<th>ILQ</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.120.1</td>
<td>192.168.120.17</td>
<td>1.000</td>
<td>1.000</td>
<td>1.00</td>
</tr>
<tr>
<td>192.168.120.3</td>
<td>192.168.120.17</td>
<td>1.000</td>
<td>1.000</td>
<td>1.00</td>
</tr>
</tbody>
</table>

The following sections discuss each segment of the table in more details.

#### 4.4.1 The links table

The links table contains the links to the neighbours on the network and an example is shown in Table 4.3.
4.4 olsrd debug output

Table 4.3: A typical link table

--- 14:28:56.80 ------------------------------------------- LINKS

<table>
<thead>
<tr>
<th>IP address</th>
<th>hyst</th>
<th>LQ</th>
<th>lost</th>
<th>total</th>
<th>NLQ</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.1</td>
<td>0.000</td>
<td>1.00</td>
<td>0</td>
<td>10</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

- **IP address** – This specifies the IP address of the interface the host is using to connect to the network.

- **hyst** – This specifies the hysteresis value for the link provided this is enabled on the interface.

- **LQ** – This is the link quality measured from the host end.

- **lost** – The number of packets lost in the last \( n \) window (\( n \) can be 10 or 20 as specified under LinkQualityWinSize).

- **total** – This is the number of packets received at that instant. This value starts from zero and is capped at the number on LinkQualityWinSize value.

- **NLQ** – This is the link quality measured from the neighbour’s end. This value can be extracted from the HELLO messages received from the neighbour. If the link to the neighbour is active and HELLO message is received this value will show. If the link goes bad and no HELLO message is received then the link is deleted from the network and the olsrd routing table.

86
• ETX – This is the link quality metric as specified by ETX. This is calculated by evaluating through this formula. \( \frac{1}{NLQ \times LQ} \).

### 4.4.2 The neighbours table

The neighbours table contains the information about the neighbours and an example is shown in Table 4.4.

**Table 4.4**: A typical neighbours table

--- 14:28:56.80 --------------------------------------- NEIGHBOURS

<table>
<thead>
<tr>
<th>IP address</th>
<th>LQ</th>
<th>NLQ</th>
<th>SYM</th>
<th>MPR</th>
<th>MPRS</th>
<th>will</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.6</td>
<td>1.000</td>
<td>1.000</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>6</td>
</tr>
</tbody>
</table>

• IP Address – This is the main address(es) of the neighbour(s).

• LQ & NLQ – This is the LQ and NLQ values of the best link that exists between the host and the neighbour. It is possible to have more than one link to a neighbour if multiple interfaces are enabled and configured on the nodes.

• SYM – This indicates if the link to the specified neighbour is symmetric. This is always determined by the *olsrd*’s link detection mechanism.

• MPR – This states whether the host has selected this neighbour to act as an MPR for the host or not.
• MPRS – The MPR Selector indicates whether the neighbour’s node has selected the host to act as an MPR for it.

• will – This is the willingness of the neighbour to act as MPR. The value can be set on the configuration file. If the option is not configured then olsrd can assign a number based on the information retrieved about the system power and update the willingness dynamically according to this information.

4.4.3 The topology table

This table displays information on the topology which olsrd has gathered from LQ TC messages. This is information on the other links on the network and the respective link quality that they possess. This is given information about other links on the network beyond the node at the next hop. Table 4.5 shows a typical topology table.

<table>
<thead>
<tr>
<th>Source IP addr</th>
<th>Dest IP addr</th>
<th>LQ</th>
<th>ILQ</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.6</td>
<td>192.168.0.2</td>
<td>1.000</td>
<td>1.000</td>
<td>1.00</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>10.0.0.5</td>
<td>1.000</td>
<td>1.000</td>
<td>1.00</td>
</tr>
</tbody>
</table>

• Source IP addr - This is the node that is reporting the link.
4.4 olsrd debug output

- Dest IP addr – This is the node that the source node is reporting the link to.

- LQ – This is the link quality calculated from the source node. This means that from the neighbour’s node it will be the NLQ to this node.

- ILQ – This is termed Inverse Link Quality. It is the numerically the same as NLQ as explained above. The convention is to call it NLQ reporting links to neighbour and call it ILQ when reporting links to the destination node in the topology table.

- ETX – This is value of the ETX metric for the link. The value is calculated in the same manner as under the link table, namely \( \frac{1}{NLQ \times LQ} \).

4.4.4 Dijkstra’s table

This table outlines the best routes that olsrd estimates for each destination that exists on the network based on the topology. A typical Dijkstra’s table is shown in Table 4.6.

Table 4.6: A typical Dijkstra’s table

--- 14:28:56.80 --------------------------------------- DIJKSTRA’S

10.0.0.6:1.00 (one-hop)
10.0.0.5:2.00 <- 10.0.0.6:1.00 (one-hop)
4.5 Experimentation measuring tools

The IP address on the left given on each line is the destination of a route. Then the remaining IP addresses specify the nodes or hops on the path between the host and the destination address. This means that moving from left to right, the packets on the network move from the destination (hop by hop) to the node that is directly connected to the host. From the table above, the host is connected to two destinations. 10.0.0.6 is directly connected and this is called its neighbour. The link to 10.0.0.5 is connected via 10.0.0.6 that the host has identified as a neighbour already. There a single hop to the first destination and two hops to the second destination.

The number after the IP addresses of the destination is the ETX of the link up to that point. ETX for 10.0.0.6 is 1.00 while the cumulative ETX for the path to 10.0.0.5 is 2.00 (1.00 + 1.00) this translates that the ETX between each of the two hops are 1.00. It also shows that 10.0.0.5 is not a neighbour of the host.

This table is updated every time the TC message is received. If any link on the network fails then the link will be removed from the table. Then Dijkstra’s algorithm will attempt to recalculate the shortest path based on the metric and re-route the packets through another path. This is the same if the quality of the link drops, then the algorithm will attempt to recalculate the route to ensure that the best route is chosen for the transmission of the traffic on the network.

4.5 Experimentation measuring tools

After the development of the plug-in program and before the implementation of the extension on the olsrd, some fundamental tests were carried out to investigate the characteristics of RSSI and its suitability for the proposed solutions. These were done by running the test program on one of the nodes and taking the measurement of the other node. The output data was sent to a text file on the measuring machine and the data was then analysed with Matlab.
4.6 Implementation Issues

At the latter part of this research, another open source network monitoring tool was used. This is called JPerf and it can be downloaded from the Source Forge website\(^1\). It is a network quality measuring tool written in Java and associated with IPerf (IPerf, 2011). This is particularly useful in measuring TCP and User Datagram Protocol (UDP) throughput and it has an incorporated GUI interface for easier configuration.

4.6 Implementation Issues

The main implementation issue with the test-beds is the chipset embedded on the wireless cards. The MadWifi driver operation depends on proprietary Hardware Abstraction Layer (HAL). The commands only work on Atheros chipsets. It was quite challenging to set up the second test-bed as most D-Link on sale at that time came with Ralink chipset\(^2\) instead of Atheros chipset\(^3\). Unfortunately, the chipset version is not always written on the box on the shelves.

MadWifi Project provides a compatibility list which are updated from time to time. This list includes different wireless card manufacturers and tries to match the chipset with every product version in the market. The list can be found on the compatibility\(^4\) section of MadWifi website.

4.7 Experimental Environments

This section describes the classification of the environments for the experiments. This includes the location of the wireless nodes, the floor-plans and pictures of environments where the experiments were carried out. The effects

---
\(^1\)http://code.google.com/p/xjperf/downloads/list
\(^2\)http://www.ralinktech.com/
\(^3\)http://www.atheros.com/
\(^4\)http://madwifi-project.org/wiki/Compatibility
of these environments on the measurements are significant and it is imperative to consider these in analysis of the measurement data.

In chapter 5, there are series of measurements and comparison of measurements at different times of the day. This is to investigate the effect of movements or multi-path on the wireless channels and how this was estimated through the measured RSSI samples. Also, in chapter 6, the implementation of the routing metric involves some classification of different environmental variables which can be best explained by defining the conditions of the wireless channel when the measurements were taken.

### 4.7.1 Overview of the experimental environment

The majority of the experiments were carried out at D109, Park Square, Luton. This is an open-office environment with about 20 desks with computers and printers. The picture of the main office and the extension to the office are displayed in Figures 4.4(a) and 4.4(b).
4.7 Experimental Environments

4.7.2 Wireless mesh node locations

The positions of the wireless nodes inside D109 and D109a are displayed in Figure 4.5.

Figure 4.4: Pictures of D109 & D109a Park Square, Luton
4.7 Experimental Environments

(a) Node 1  (b) Node 2

(c) Node 3  (d) Node 4

(e) Node 5 (D109a)

**Figure 4.5:** Wireless mesh node locations
4.7.3 Morning measurement environment

Table 4.7 displays the classification of the environments and the time of the day when these happened. Also the number of people or human traffic within the region at that particular time. For the morning time measurement, this occurred between 0830hrs and 1030hrs in the morning. At this time, few people are yet in the office and the human movement was minimal.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Time (hrs)</th>
<th>Number of people in the Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morning</td>
<td>0830 - 1030</td>
<td>between 4 to 6 people</td>
</tr>
<tr>
<td>Afternoon</td>
<td>1400 - 1530</td>
<td>minimum of 18 people</td>
</tr>
<tr>
<td>Night</td>
<td>2030 - 2300</td>
<td>1 person</td>
</tr>
</tbody>
</table>

Table 4.7: Table showing the environment classifications for the experiments

Figure 4.6 shows the floor-plan of D109 and D109a in the morning. D109a was always empty and no movement or people there when any of the morning measurements were taken.

Figure 4.6: D109 & D109a in the morning time
4.7 Experimental Environments

4.7.4 Afternoon measurement environment

The afternoon environment is always after lunch time. This is between 2pm and 3.30pm. The majority of the measurements were taken at this time. This is because human traffic is significant at this time; there were a minimum of 18 people in the office when the afternoon measurements were taken. The environment of the afternoon time is displayed in figure 4.7.

![Figure 4.7: D109 & D109a in the afternoon time](image)

4.7.5 Night measurement environment

Night measurements are taken late in the evening between 8:30pm and 11pm. At this time, no one is in the office except the person taking the measurement, so there is no human movement in the office at this time except that of the author. The environment of the night time is displayed in figure 4.8.
4.7 Experimental Environments

4.7.6 ANTLab Test-bed

A few of the initial experiments were carried out at ANTLab Milano, Italy. Especially, the experiments where the effect of artificial interference (noise) on the received signal strength was investigated. The experiment was not repeated in Luton because the use of microwave ovens was not permitted there. This experiment is reported in chapter 5. The main office is about 12 metres × 6 metres, housing about 15 desks with desktop/laptop for research students. The floor-plan of ANTLab is displayed in figure 4.9.

Figure 4.9: Environment at ANTLab Milano for the experiments
4.8 Summary

One of the objectives of this research is to implement the solution on WMN test-beds to evaluate the performances. The open source software approach is useful in achieving this as it is free and readily available. More importantly is the fact that it can be configured to suit the design and scenarios of the user without having to obtain permission or license from the primary developer. This was one of the primary challenges the author encountered at the very beginning of this research work, as the available tools are not easily modified for research use without the intervention of the developer.

The implementation of this work was solely on test-beds. The software used was all open source. This is rather time consuming and challenging to achieve but the results obtained are practical and realistic. This fulfills one of the objectives of this study.

The implementation environments for the experiments were described in this chapter. The effects of these classifications are better appreciated in the next two chapters when the RSSI and the metric implementation is carried out. The next chapter will discuss the experimentation part of this research. This is by investigating the dynamics of RSSI under different conditions and scenarios.
Chapter 5

RSSI Experimentation

This chapter reports the experimentation of this research. The nature and characteristics of RSSI are very challenging. Hence, various measurements were carried out to study and analyse the behaviour of RSSI. All the data was obtained on implemented test-beds using wireless cards as described in the previous chapter. The study of RSSI variability using statistical analysis is discussed here. The characteristics of the function under different channel conditions are also investigated and lastly the theoretical pattern that the function follows is also investigated.

5.1 Statistical analysis of RSSI variability

In order to effectively implement RSSI as a driving metric for a WMN routing protocol, it is important to investigate the nature and the signal characteristic of RSSI. Previous works avoided the use of RSSI as a metric of RSSI because of the common belief of its randomness without actually investigating or modelling its characteristics (Vlavianos et al., 2008). To study its characteristics and dynamics, it became necessary to carry out some preliminary experimentation on RSSI. This was to make some necessary and realistic conclusions which would help in the implementation of the routing protocol and be a basis
5.1 Statistical analysis of RSSI variability

to leverage for future studies.

Some measurements were taken at different times of the day, i.e. in the morning, afternoon and night. These measurements are then analysed to investigate the characteristics of the measured values. The mean, standard deviation, frequency and the auto-correlation of the data were then calculated and analysed.

The standard deviation of a set of data is the measure of the variability of the data and is expressed as:

\[
\sigma_N = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2}
\]  

(5.1)

where \( \bar{x} \) is the mean value of the data set.

Also, auto-correlation which describes how fast a signal de-correlates can be used to ascertain the randomness of a data set. For a time series or measurement data like RSSI, the randomness of the data set should be tested. This is achieved by computing the general dependency of a data \( x(t) \) with its value at a short time later \( x(t + \tau) \). This can be expressed as thus:

\[
\rho_x(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} [x(t) - \bar{x}] \cdot [x(t + \tau) - \bar{x}] \, dt
\]  

(5.2)

where \( \tau \) is the time constant which is the time lag between the time of the measured value and the time the value was correlated with itself.

5.1.1 Morning measurement data

Figures 5.1 to 5.3 show the measurement data of RSSI values in the laboratory in the morning time. At this time, the laboratory was almost empty and there is less movement around the laboratory which could cause interference due to multi-path. The histogram shows the frequency of the measured value. From the diagram, it was observed that the modal RSSI value for this reading is -52dBm. The auto-correlation plot is also shown; this shows that at point 0, the
5.1 Statistical analysis of RSSI variability

auto-correlation of the data is 1. This means that there is strong correlation between the data, and the samples are not simply random samples.

Figure 5.1: RSSI measurement taken at morning time

Figure 5.2: Histogram of RSSI data at morning time

Table 5.1 displays the statistical values of the data which include the mean, mode, median etc.
5.1 Statistical analysis of RSSI variability

![Graph showing autocorrelation of RSSI data at morning time.](image)

**Figure 5.3:** Autocorrelation of RSSI data at morning time

**Table 5.1:** Statistical values of RSSI data at morning time

<table>
<thead>
<tr>
<th>Statistical Variables</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>-55</td>
</tr>
<tr>
<td>max</td>
<td>-46</td>
</tr>
<tr>
<td>mean</td>
<td>-51.6</td>
</tr>
<tr>
<td>median</td>
<td>-52</td>
</tr>
<tr>
<td>mode</td>
<td>-52</td>
</tr>
<tr>
<td>std</td>
<td>1.517</td>
</tr>
<tr>
<td>range</td>
<td>9</td>
</tr>
</tbody>
</table>

5.1.2 Afternoon measurement data

Figures 5.4 to 5.6 show the afternoon version of the measurement data. This histogram and the auto-correlation plot is also shown.

Table 5.2 shows the statistical measures of the measurement data and this tallies with the histogram where the modal measured value shows -59dBm.
5.1 Statistical analysis of RSSI variability

Figure 5.4: RSSI measurement data taken at afternoon time

Figure 5.5: Histogram of RSSI data at afternoon time
5.1 Statistical analysis of RSSI variability

Figure 5.6: Auto-correlation of RSSI data at the afternoon time

Table 5.2: Statistical values of RSSI data at afternoon time

<table>
<thead>
<tr>
<th>Statistical variables</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>-73</td>
</tr>
<tr>
<td>max</td>
<td>-55</td>
</tr>
<tr>
<td>mean</td>
<td>-61.32</td>
</tr>
<tr>
<td>median</td>
<td>-61</td>
</tr>
<tr>
<td>mode</td>
<td>-59</td>
</tr>
<tr>
<td>std</td>
<td>2.798</td>
</tr>
<tr>
<td>range</td>
<td>18</td>
</tr>
</tbody>
</table>

5.1.3 Night measurement data

The night measurement data is displayed in Figures 5.7 to 5.9. The histogram confirms that the modal measured value of the RSSI is -47dBm as shown in Table 5.3. The auto-correlation plots also confirm the non-randomness of the data set.
5.1 Statistical analysis of RSSI variability

![Figure 5.7: RSSI measurement data taken at night time](image)

![Figure 5.8: Histogram of RSSI measurement at night time](image)
5.1 Statistical analysis of RSSI variability

![Figure 5.9: Auto-correlation of RSSI measurement at night time](image)

Table 5.3 shows the statistical measures of the night measurement data and this tallies with the histogram where the modal measured value shows -47dBm.

**Table 5.3:** Statistical values of RSSI data at night time

<table>
<thead>
<tr>
<th>Statistical variables</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>-50</td>
</tr>
<tr>
<td>max</td>
<td>-42</td>
</tr>
<tr>
<td>mean</td>
<td>-47.07</td>
</tr>
<tr>
<td>median</td>
<td>-47</td>
</tr>
<tr>
<td>mode</td>
<td>-47</td>
</tr>
<tr>
<td>std</td>
<td>1.31</td>
</tr>
<tr>
<td>range</td>
<td>8</td>
</tr>
</tbody>
</table>

The main conclusion in the first part of this experiment is the randomness of the RSSI samples. The graphs in Figures 5.3, 5.6 and 5.9 have auto-correlation ($\rho(\Delta t)$) on their y-axis and time-lags on their x-axis ($\Delta t$). For all the three environmental classifications (morning, afternoon and night), the
5.2 Comparison of statistical data for the different environmental measurements

auto-correlation is 1 at time 0. Hence, it can be concluded that none of the samples are random. The three measurements are compared in the next section to observe the degree of auto-correlation due to the difference in the environmental conditions.

5.2 Comparison of statistical data for the different environmental measurements

Having confirmed the non-randomness of the data set for the different times of the day, the next step was to compare the characteristic of the RSSI distribution. This was to investigate the changes in the distribution as activities started to build up in the lab. The laboratory houses about 18 researchers (post-doctoral and PhD students) and it was always busy in the afternoon time except at break times. There were movements up and down and this had interference effects on the measured data as a result of multi-path.

Table 5.4 and Figure 5.10 highlight the range, standard deviation and the mean values of the morning, afternoon and night measurement data. Range and standard deviation statistically measure the spread of the data. Otherwise, the mean statistically predicts the expected value of a random variable or the measure of central location of a data.

The range and standard deviation are in agreement in this data set and are least in the night next to the morning and the largest in the afternoon. The main reason for this is the interference level in the afternoon which obviously had an effect on the behaviour of the received signal strength. In the morning the activity in the laboratory is at minimum, whereas there is practically no movement at all in the night. The night range and standard deviation is slightly less than that of the morning for this reason.

The mean value is highest in the night when the transmission channel is less congested and most wireless routers are already switched off. In the morning,
the signal is still high but slightly lower than the night signal. The afternoon mean is the lowest which further confirms the effect of interference and multi-path on the transmission. Figure 5.11 shows the RSSI for the different measurements and it is evidenced in the figure that the signal is best at night followed by morning and least in the afternoon.

Table 5.4: Table comparing the range, standard deviation and mean values of measured RSSI data

<table>
<thead>
<tr>
<th>Time of the day</th>
<th>Range (dBm)</th>
<th>Std (dBm)</th>
<th>Mean (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morning</td>
<td>9</td>
<td>1.517</td>
<td>-51.6</td>
</tr>
<tr>
<td>Afternoon</td>
<td>18</td>
<td>2.798</td>
<td>-61.32</td>
</tr>
<tr>
<td>Night</td>
<td>8</td>
<td>1.31</td>
<td>-47.07</td>
</tr>
</tbody>
</table>

Figure 5.10: Range, standard deviation and mean values of RSSI measurement data at different times of the day
5.2 Comparison of statistical data for the different environmental measurements

Figure 5.11: The comparison of measurement data showing the differences in signal levels at different times of the day

The graphs in Figure 5.12 show the auto-correlation function for the measurement data. In accordance with the observation and analysis from other statistical variable, the signal has the strongest auto-correlation at night and a slightly stronger one in the morning. Although the auto-correlation in the afternoon did not depict randomness, but this is weaker compared to that of the corresponding measurement in the morning and at night.
5.3 PDF of the measurement data

Figure 5.12: Comparison of measurement data showing the auto-correlation functions of the data

More results are documented in appendix C. These include the data from different experimental environment, with their autocorrelation and histograms.

5.3 Probability Density Function (PDF) of the measurement data

It is important at this stage to characterise the RSSI function. This is to observe how the signal behaves and predict the trend of the signal under various conditions. This is needed to understand how to adapt the parameters for an optimum performance.

The PDF of an absolute continuous variable as described in probability theory is the relative chance of the variable to occur at a given point in time. A common example in applied engineering is the normal or Gaussian distribution or density function. This is expressed as:

\[ f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]  \hspace{1cm} (5.3)
5.3 PDF of the measurement data

where $\mu$ is the mean, $\sigma^2$ the variance and $e$ the Euler number.

Gaussian PDF is associated to a “bell-shaped” function, or bell curve. The parameters of the mean and the variance determine the shape of this function. The mean value determines the location of the peak while variance measures the width of the distribution.

Gaussian PDF for the measured RSSI values was also generated. This was generated by MatLab to superimpose the theoretical PDF on the histogram of the measured data, purposely to observe the trend of the data. From Figure 5.13, the three measurements fit a theoretical Gaussian PDF though the evening measurement seems to look more like a Rayleigh distribution. Although there are few points outside the traditional bell shape, they can be better processed by passing the data through a low pass filter. This filtering process will be further investigated in the next chapter.

The PDF shows the highest signal and highest frequency for the night data followed by the morning data. The signal is strongest at night as shown in the figure with the highest peak. Afternoon data is somehow spread with the largest value of variance (square of the standard deviation). The signal is relatively weak in the afternoon and more dispersed.
5.4 Noise and transmit power effects on the measured RSSI

Further study is required to conclude on the theoretical distributions trend that RSSI samples follow under different conditions. In this experiments, the majority of the samples follow a Gaussian distribution while some follow a Rayleigh distribution. More experimental data are documented in appendix C with Gaussian PDF superimposed on the experimental results.

5.4 Noise and transmit power effects on the measured RSSI

The characteristics of measured RSSI data was further investigated by taking measurements of a remote wireless node by increasing the transmit power. These were carried out at three main levels, i.e. 1dBm, 7dBm and 16dBm. This was to investigate the effects of increased transmit power on the received signal strength. The wireless card MadWifi driver enabled the manual change of the \texttt{txpower} function of the card through the MadWifi \texttt{iwconfig} command.
5.4 Noise and transmit power effects on the measured RSSI

Two parameters were controlled simultaneously in these sets of measurements. The effect of transmit power increments and the effect of noise on the measured data. Noise in form of signal interference was generated using a 1KW microwave in the lab. The microwave was placed in such a way that the microwave radiation from the device will obstruct or distort the signal between the two points. The wireless node under observation (node 2) was beside the microwave and the remote laptop (node 1) taking the measurements was placed at the other end of the laboratory. The measurements were taken at night time due to the possible effect of continuous microwave exposure on students working near the microwave oven.

![Figure 5.14: Artificial interference measurement environment on node 2](image)

The experiment was repeated without the microwave radiation in order to set up a control scenario. The power levels were also adjusted to obtain readings which can be compared with the artificial interference experiment. Figure 5.15 shows the trend of the measured data while Figure 5.16 shows the distribution and the PDF of the data. It can be observed that increments in the transmit power correspond to increments in the received signal strength but the ratio is not linear. This is expected because power radiates in an exponential pattern.
5.4 Noise and transmit power effects on the measured RSSI

![Graphs of measurement data of a node under artificial interference with different transmit power settings](image1)

**Figure 5.15:** Graphs of measurement data of a node under artificial interference with different transmit power settings

![Histogram of measurement data of a node under artificial interference with different transmit power settings](image2)

**Figure 5.16:** Histogram of measurement data of a node under artificial interference with different transmit power settings
5.4 Noise and transmit power effects on the measured RSSI

Figure 5.17 and Figure 5.18 show the trend of the measured data and the frequency of the data respectively. Just like the preceding experiment, a slight increment was observed accordingly with the increment in the transmit power. This shows that increment in the transmit power will increase the signal strength level but in a diminishing pattern.

![Graphs of measurement data of a node without artificial interference with different transmit power settings](image)

**Figure 5.17:** Graphs of measurement data of a node without artificial interference with different transmit power settings
5.4 Noise and transmit power effects on the measured RSSI

Table 5.5 and Figure 5.19 show further statistical derivations from the measurement data under the artificial interference. Although the mean values increased slightly, the standard deviation increased also. If the signal is becoming stronger due to the increased power the standard deviation is expected to appreciably reduce in value. The range values did not follow any particular pattern as observed from the table. This needs further investigation to clarify the reasons behind the behaviour.

Table 5.5: Table comparing the mean, standard deviation and range values of different power levels under artificial interference

<table>
<thead>
<tr>
<th>Transmit Power Level</th>
<th>Mean</th>
<th>Std</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1dBm</td>
<td>27.40</td>
<td>2.506</td>
<td>11</td>
</tr>
<tr>
<td>7dBm</td>
<td>32.73</td>
<td>2.797</td>
<td>10</td>
</tr>
<tr>
<td>16dBm</td>
<td>35.83</td>
<td>3.216</td>
<td>17</td>
</tr>
</tbody>
</table>
5.4 Noise and transmit power effects on the measured RSSI

![Graphs showing mean, standard deviation, and range values of RSSI measurement data at different transmit power levels under artificial interference.](image)

Figure 5.19: Mean, standard deviation and range values of RSSI measurement data at different transmit power levels under artificial interference
5.4 Noise and transmit power effects on the measured RSSI

For the control experimental setup, the mean value increased as the power level is increased. This is not in a linear or logarithmical order as expected but in the same manner as the previous experiment. This is expressed in Table 5.6 and Figure 5.20. The standard deviation and the range values did not follow a particular order. Hence, it can be concluded that increase in the power level brought appreciable increase in the signal strength. However, the effect noise and transmit power levels on the variability of the signal which the standard deviation and the range values depicts need further investigation.

Table 5.6: Table comparing the mean, standard deviation and range values of different power levels without artificial interference

<table>
<thead>
<tr>
<th>Transmit Power Level</th>
<th>Mean</th>
<th>Std</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1dBm</td>
<td>22.88</td>
<td>0.613</td>
<td>2</td>
</tr>
<tr>
<td>7dBm</td>
<td>26.03</td>
<td>1.178</td>
<td>4</td>
</tr>
<tr>
<td>16dBm</td>
<td>33.93</td>
<td>0.710</td>
<td>2</td>
</tr>
</tbody>
</table>
5.4 Noise and transmit power effects on the measured RSSI

![Graphs showing mean, standard deviation, and range comparison of RSSI measurements at different transmit power levels without artificial interference.](image)

(a) Mean comparison  (b) Std comparison  (c) Range Comparison

**Figure 5.20:** Mean, standard deviation and range values of RSSI measurement data at different transmit power levels without artificial interference

Lastly, figure 5.21 combines the effects of transmit power increment on the RSSI by observing the range and the standard deviation for both interference condition and the control experiment.
5.5 Summary

The characteristics of the measured RSSI data were critically analysed in this chapter. The variability of the function which is challenging to model has shown a non-randomness characteristic through the auto-correlation analysis of the data. Statistical analysis of a random process like RSSI helped in analysing the measured data at different times of the day under different environmental conditions. Results show that RSSI signal levels are correlated with the link conditions and the variability are equally a function of the experimental environment. The multi-path effect of a busy environment has a negative effect on the received signal strength.

The effects of noise and transmit power levels on the measurement data were also observed. Increments in the power level increases the signal level of the link. The effect of noise on the measure of dispersion on the data did not show a strong correlation. Hence this needs further investigation.

More results on the effects of noise and transmit power effect on the RSSI are documented in appendix C.

![Graphs showing range and standard deviation comparison for RSSI samples without and with artificial interference.](image)

**Figure 5.21:** Range and standard deviation comparison for RSSI samples without and with artificial interference
The characteristic of the data has been compared and the PDF using the Gaussian distribution predicts the trend of the function. The majority of the samples fit the superimposed theoretical Gaussian PDF while some seem more like a Rayleigh distribution. The experimental results show how the empirical PDF and the theoretical Gaussian PDF matches in most cases. However, it was observed that there are few points outside the normal bell shape. This could be as a result of the variability which can be controlled to a certain extend by filtering the measurement data. This will be addressed at the next chapter where the low pass filtering of the RSSI signal will be investigated and implemented.
Chapter 6

rETT Implementation

The necessary experiments to study the characteristics of RSSI were described in the last chapter. Some of the observations and the conclusions will useful in the implementation of the proposed routing metric covered in this chapter. Dealing with the variability of the RSSI samples and the implementation of this function into the OLSR routing protocol are key to the success of this research. Hence, the RSSI filtering process introduced in chapter 3 (under the rETT metric algorithm) is implemented at this stage. Consequently, the routing protocol based on rETT metric is implemented also and their network performances and comparison with other related metrics investigated. Finally, the need to optimise the performances of the proposed metric is considered at the concluding stage of this chapter.

6.1 Implementing RSSI Filter

This section describes the experimental results obtained on the filtering process and the relevance of this data to the set objectives. The first and fundamental task was to justify the use of RSSI information to drive the metric of OLSR routing protocol; this was challenging because of the variability of the RSSI information measured at the neighbouring nodes.
Some of the experimental results in chapter five show the nature of the variability of RSSI samples. Especially when superimposed with the Gaussian PDF, it was observed that some of the plots are not normally distributed. The fundamental of RSSI variability was introduced in chapter two. A typical RSSI sample is highlighted in figure 6.1. This shows a relatively strong RSSI signal expressing the variability of the sample.

Due to the variability of the RSSI, the raw value of the RSSI cannot be directly used as a metric of any routing protocol. The reason being that every change in the RSSI value triggers a re-computation of the routing protocol and hence makes the routing process very unstable. Also the routing table of the nodes is updated and this consumes some network resources and could degrade the level of network performance. The main challenge is to choose a filter and a filter constant which is not only steady in operation but also able to reflect when the level of the received signal strength drops or increases, i.e. finding a realistic balance between stability and sensitivity.

The details of the algorithm are documented in chapter 3 and the experiments to investigate the filtering aspect of the algorithm are documented here. The IOCTL calls helped in providing the RSSI information as measured by the

![Figure 6.1: RSSI variability](image-url)
wireless cards based on Atheros Communications chipsets. These values are hence stored in memory and updated every second. The average value represents the exponentially moving average values of the RSSI samples. These are returned to the program to transform the value to the appropriate rate information based on the sensitivity limit of the chipset. The rate information based on the sensitivity limits was then used in driving the metric of the RSSI-aware OLSR to compute the routes to be chosen for transmission of packets on a WMN.

6.2 RSSI Filtering Experiments

The functionality and the response of the filter were listed after implementing the codes for the RSSI filter. Various experiments were conducted to ascertain the best filter constant to use, to produce an optimal result that is stable enough for the computation of the routing protocol. This should also be sensitive enough to detect any significant changes in the signal strength level of the nodes in the WMN.

The dynamics of $\alpha$ value which is the filter constant is significant to the performance of rETT. In chapter 3, the design of the rETT and the dynamics of the metric were proposed. It was mentioned that the benefit of rETT is best appreciated when the wireless channel is poor or the signal signal strength is low. In this case, the metric is able to compute the link quality for the network path and choose better links to form the topology of the network. To appreciate this better, the $rssi_to_rate$ parameter is added to the measured RSSI samples and the RSSI moving average.

Any change in the transformed rate value is capable of triggering a change in the topology of the network as this is used in the computation of the metric. If this is higher or lower, the routing algorithm will take this into consideration and assign a cost value to it which the algorithm will evaluate in constructing the routing table for the WMN at every instance of topology change.
This is investigated under two scenarios. The first one is when the link is subject to multi-path effect (i.e. measurements were taken in the afternoon) and the distance between the node under observation and the measuring node are wide apart and not within Line of Sight (LOS). While the second one is when the nodes are close together and within the LOS and the movement in the environment is minimal (measurements were taken in the morning).

- **Scenario One**

Figure 6.2 shows the positions of the two nodes in the environment when the measurements were taken.

![Figure 6.2: Layout of experimental setup for filtering experiment - scenario 1](image)

The following are the results obtained when investigating the effects of different filter constants on the RSSI.

Figure 6.3 shows the trend of the rate value which is computed from the sensitivity limits transformation (rssi to rate) of the filtered RSSI values. From this figure, it can be observed that the rate value only changed twice throughout the 60 seconds of observation. It changed at $t = 4s$ from 12Mbps to 18Mbps and also changed at $t = 44s$ from 18Mbps to 24Mbps. This means that the
6.2 RSSI Filtering Experiments

topology is likely to be changed twice in that minute provided all other link conditions on the network remain the same.

![RSSI Filter with alpha = 0.01](image1)

**Figure 6.3:** RSSI Filter with alpha = 0.01

With the $\alpha$ value increasing to 0.125, the effect on the rate value is more than the previous setting. The changes increased from 2 to 4 and this happened within $20s < t < 31s$.

![RSSI Filter with alpha = 0.125](image2)

**Figure 6.4:** RSSI Filter with alpha = 0.125

The rate value when $\alpha$ was changed to 0.25 was much more significant. The
rate value changed 19 times in one minute. This can be considered to be a better representation of the link condition but it comes with a side effect. The implication for the routing process is that the changes will trigger re-computation of the routing table which may eventually changed the topology of the network 19 times in one minute.

![Graph](image)

**Figure 6.5:** RSSI Filter with alpha = 0.25

The same effect was experienced when the $\alpha$ value was increased to 0.375. The rate value changed 25 times in one minute. This happens on average every 2.4 seconds. It should be noted that every re-computation of the topology will require TC information to be flooded in the network. This is a bottleneck to the network resources and it should be controlled.
The graphs in Figures 6.3 to 6.6 show the measured RSSI values, the computed RSSI-MA values and the corresponding rate values in Mbps. The effect of the alpha value or the filtering constant can be seen on the targeted rate information. It should be noted that this rate information \( r \) is to be incorporated as part of the metric to compute the \( rETT \) of the OLSR.

When \( \alpha = 0.01 \), the rate information is quite stable, this is good for the routing protocol. The topology will also be relatively stable but less sensitive to changes in the link condition. The situation is slightly different with \( \alpha = 0.125 \), the rate information for the link is more sensitive to the changes but can still be considered as quite stable. The situation follows the same trend for \( \alpha = 0.25 \) and for \( \alpha = 0.375 \). The rate information was somehow sporadic as the value of alpha increases. This seems to be too sensitive and the routing protocol will have to trigger many topology re-computations, which is not ideal for routing protocol. From inspection of all the tested filter constants, \( \alpha = 0.25 \) seems to be an appropriate value. Hence, this value was used in the implementation of the RSSI-aware routing protocol for the OLSR at this stage of the research.
6.2 RSSI Filtering Experiments

The chosen moving average filter performed well in filtering the RSSI in this situation and it dealt with the variability of the value effectively.

- **Scenario Two**

In this scenario, the nodes are close together, the measurements were taken in the morning and there is a LOS between nodes 1 and 2.

![Layout of experimental setup for filtering experiment scenario - 2](image)

**Figure 6.7:** Layout of experimental setup for filtering experiment scenario - 2

Figures 6.8 show the effects of $\alpha$ values on the RSSI samples. Because the link is good, changes in the RSSI values did not change the value of $\text{rssi\_to\_rate}$ which is constant throughout the steps of changing $\alpha$ values in the experiment. This in effect means the metric under this scenario is insensitive to the RSSI values and will only depend on ETX value or at best based the topology formation on hop counts. However, the effect of the constants on the moving average values can still be appreciated. In figure 6.8(a), the moving average value is insensitive to RSSI but the response was improved with incrementing in the $\alpha$ values from 0.125 to 0.5. This can be observed in figures 6.8(b) to 6.8(e).
6.2 RSSI Filtering Experiments

Figure 6.8: RSSI Filter with different values of alpha - 3

The next phase of experiments shows the implementation of the RSSI-aware
6.3 Routing Performances

After implementing the rETT metric on OLSR, the performance of the proposed metric is investigated on the test-bed. This is in order to analyse the performance and compare it with the traditional metric based on hop counts and the ETX link quality metric. These are the closely related metric functions that could justify the efficiency of the proposed metric. Also, this control metric can be implemented on the same test-bed and hence the performances can be effectively compared. The following sections report the finding of the metric performances with hop counts and ETX.

6.3.1 Hop Counts

The next experiment shows the performance of rETT compared to the traditional hop count metric. It should be noted that classical OLSR will automatically choose single hop provided the link is reachable. This is irrespective of the state of the link. But the RSSI-aware OLSR will consider also the quality of the link through the received signal strength to compute the routes for the network. This is investigated in the following experiments.

- Scenario One

Considering a four-node WMN scenario where packets could go from node 1 directly to node 4 or go via node 3. The option of going directly to node 3 is made possible because the HELLO message from node 1 can be acknowledged by node 4.

Figure 6.9 and figure 6.13 show the network arrangement of the WMN and the olsr debug output confirming the network topology respectively. Node 1
connects directly to node 2 and also there exist a wireless connection to node 3 and finally to node 4 via node 3. Figure 6.10 to Figure 6.12 show the traceroute from node 1 to other nodes to confirm the network connectivity among them. This is in agreement with the routing table which shows olsr debug output in Figure 6.13.
Figure 6.9: Experimental Test-bed 1 (4 Wireless Mesh Nodes)
6.3 Routing Performances

Figure 6.10: Traceroute output between from Node 1 to Node 4

Figure 6.11: Traceroute output between from Node 1 to Node 2
6.3 Routing Performances

Figure 6.12: Traceroute output between from Node 1 to Node 3

The routing table from node 1 has the data in Figure 6.13. This shows the routing table and also the link information between the three other nodes taking reference from node 1 (172.16.3.1).

Figure 6.13: OLSR daemon debug output 1
6.3 Routing Performances

The routing table in Figure 6.13 shows that node 1 (172.16.3.1) has the best RSSI to node 3 (172.16.4.1). Hence, it was able to establish a link of 54Mbps while node 4 (172.16.5.1) which is farther away with RSSI of 1 and a lower link rate of 6Mbps. Node 1 is only able to reach node 4 via node 3 if the link quality is considered in the metric. By shortest path metric, node 1 will attempt to connect to node 4 directly and if such a link is established the throughput on the link will be very low due to the bad state of the link.

In this scenario, ETX metric was also able to differentiate between the links and gave a good judgement regarding the state of the three established links. But more importantly, it was observed that $r\text{ETT}$ performs better than the classical OLSR routing metric based on hop count.

- Scenario Two

Another experiment was conducted to validate the conclusion of the experiment in scenario 1. The environment was the same as that of scenario 1. The network settings were retained but the link conditions were altered to confirm if the $r\text{ETT}$ metric is able to detect the changes and adapt the routing accordingly. This is also compared with the traditional approach by hop counts. Figures 6.14 and 6.15 show the topology and routing table for this scenario. From the $\text{olsr}$ debug output, the new topology arrangement is shown.

In this scenario, the transmit power of node 2 (172.16.2.1) was reduced by 3dBm and the RSSI dropped from 16 to 12. In the previous scenario, node 2 was directly connected to node 1. But due to the drop in received power, the link between node 1 and node 2 is no longer sufficient. Then, the RSSI-aware metric redirected the route through node 3 (172.16.4.1) which has a better link quality to node 1 and also a better link quality to node 2.

The ETX for the links from node 1 and node 2 to node 3 shows that ETX equals 1.00 for the two links and hence assume the same link state for both of them. Actually, both links have a difference of 10 (22-12) in RSSI and a difference of 39Mbps (54-15) in link rate. Hence to send packets to node 2
6.3 Routing Performances

Figure 6.14: Experimental Test-bed 2 (4 Wireless Mesh Nodes)

Figure 6.15: OLSR daemon debug output 2
or node 4, the packet needs to go via node 3, which is the best link in this scenario.

### 6.3.2 ETX Metric

In another mesh network scenario consisting of 5 Wireless Mesh Nodes, the following experiment shows the deficiencies of using routing metric ETX alone on a WMN. The other two scenarios focused on the hop counts while this one focuses on the ETX metric. Furthermore, it is necessary to establish the intelligence of RSSI-aware OLSR over the OLSR Link Quality metric by ETX.

Figure 6.16 and Figure 6.17 show the topology and the \textit{olsr} debug output for the WMN. On the link table, node 1 (172.16.1.1) has equal ETX on the direct links to nodes 2, 4 and 5. In this situation, ETX metric can choose any link at random based on the calculated ETX values as shown by the ETX metric information. Hence, \textit{olsrd} using ETX metric only will assume that link qualities are equal for the 3 links. The explanation below confirms that RSSI aware information was consistent in choosing the best path and hence will achieve a better performance for the network.
6.3 Routing Performances

Figure 6.16: Experimental Test-bed 3 (5 Wireless Mesh Nodes)
In this scenario, there are 5 nodes; all readings are from node 1 (172.16.1.1).

The transmit power of node 1 and node 5 was deliberately set at maximum (19dBm) for the test-bed wireless card. Also, the path between them was free from interference. These two nodes have the longest physical distance between them but in terms of link quality, the link between them is the best. The RSSI between them is the highest (16) and the link rate is also the highest (42Mbps). In terms of the ETX for the links, the metric between them and node 1 is coincidentally 1.11, which means ETX will assume the same link quality for the links between nodes 1 to node 2, node 1 to node 4 and node 1 to node 5. In reality, the RSSI between node 1 and node 5 is the best (16) while node 1 and node 2 returned half of this value (8). By shortest path and ETX, there is one hop between node 1 and node 2 but by RSSI aware ETT the topology prefers to reach node 2 via node 5.

Looking at the ETT column under the link table, the time required to transmit packets between the nodes also justify this claim. ETT between node 1 and node 2 is much higher (2.227s) than the ETT between node 1 and node 5 which is 0.476s. Hence, the routing performance of the proposed rETT is justifiably

---

**Figure 6.17:** OLSR daemon debug output 3

---
better than the traditional hop count and the ETX link quality metrics.

6.4 Optimizing the routing performance

At the first phase of this implementation, the alpha value used was 0.25. This value was chosen based on inspection of the filter performances when the experiment was carried out, as described in section 6.2. However, a more empirical approach is needed to determine the optimum value of the filter constant that will yield the best network performance. Taking into consideration the sensitivity of the network to link changes and the stability of the routing process. This is the focus of this section.

As mentioned earlier, the RSSI value was filtered by using an exponentially moving average as expressed in equation (3.5). The filter constant, alpha ($\alpha$) determines the stability and sensitivity of the routing protocol. From the equation, a low alpha value means that more of the previous values are used in the iteration to determining the RSSI value. Hence, the resulting average value that will be used in the routing protocol has more of the past value in the memory and less of the new value. The issue with this is that, if there is a sudden change in the received signal strength, the routing protocol would not be updated immediately. It will take a time lapse before the change is reflected in the computation of the routing protocol. While a high alpha constant, will be more sensitive to any instantaneous change and has less of the previous value in the memory for the iteration of the average value. This implies that a typical high value will be more sensitive to the link fluctuations but less stable routing process.

To effectively optimise the performance of \( rETT \) routing metric, there is a need to take a second look at the metric proposed. The metric is an estimation of the time taken for packets to transverse from the source to the destination and this is expressed in seconds. The \( rETT \) of every link is estimated and the \( rETT \) of the path is the summation of the respective \( rETT \) of the links.
6.4 Optimizing the routing performance

involved in the path. Therefore, the routing protocol adapts the path selection by selecting the path with the minimum path metric ($rETT_p$).

While the focus of the routing performance is to achieve a link quality metric that will be sensitive to the changes in the link conditions, it is equally important to factor in the effect of these changes on the routing process. Too many changes will require more control messages and these consume bandwidth. Figure 6.18 shows the setup for the performance optimisation experiment. This involved measuring the end-to-end throughput from node 5 to node 1 using different alpha values in turn. This was carried out using the *jperf* measuring tool.

![Experimental Test-bed 3 (5 Wireless Mesh Nodes)](image)

**Figure 6.18:** Experimental Test-bed 3 (5 Wireless Mesh Nodes)

Throughput measurements were conducted using TCP packets for 30 seconds. This was repeated ten times and the average of the throughputs between these links were recorded. The trends of experiments were again recorded for values of alpha from 0.05 up to 0.25 (ten steps). This was implemented by changing the alpha value on *src/main.c* and after every change the routing files were recompiled to effect the changes in the *olsrd* application. This exercise was
carried out on all nodes on the network. It is important that all nodes on the network run similar olsr daemon for the routing process to be effective.

Values above 0.25 were not considered because at this level the sensitivity of the routing will be too high and the routing protocol may find it difficult to converge. The path keeps changing almost every second or any time the TC message is received. This will introduce another complexity to the operation and hence they were exempted from the optimum test.

Table 6.1 and Figure 6.19 display the findings of the experiments and it was observed that when the value of alpha was set to 0.125, the network throughput was the best on the average. Hence the final value of alpha for the routing protocol was left at 0.125.

![Graph showing Determination of best alpha value](image)

**Figure 6.19:** Routing performances on the network with different alpha values
Table 6.1: Table showing throughput analysis on the network with different values of alpha

<table>
<thead>
<tr>
<th>α</th>
<th>0.025</th>
<th>0.05</th>
<th>0.075</th>
<th>0.1</th>
<th>0.125</th>
<th>0.15</th>
<th>0.175</th>
<th>0.2</th>
<th>0.225</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No of runs</td>
<td>Throughput (Mbps)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.38</td>
<td>1.60</td>
<td>1.63</td>
<td>1.66</td>
<td>2.44</td>
<td>1.56</td>
<td>1.69</td>
<td>1.83</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.35</td>
<td>2.09</td>
<td>1.97</td>
<td>1.80</td>
<td>2.57</td>
<td>1.67</td>
<td>1.53</td>
<td>1.88</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.36</td>
<td>2.05</td>
<td>2.15</td>
<td>1.81</td>
<td>1.85</td>
<td>1.76</td>
<td>1.26</td>
<td>1.79</td>
<td>1.87</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.77</td>
<td>1.88</td>
<td>1.39</td>
<td>1.54</td>
<td>3.32</td>
<td>1.68</td>
<td>1.97</td>
<td>1.87</td>
<td>1.47</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.07</td>
<td>1.62</td>
<td>1.41</td>
<td>2.00</td>
<td>1.72</td>
<td>1.65</td>
<td>2.09</td>
<td>2.16</td>
<td>2.15</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.00</td>
<td>2.07</td>
<td>1.47</td>
<td>2.49</td>
<td>2.57</td>
<td>1.35</td>
<td>1.75</td>
<td>2.09</td>
<td>2.05</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1.01</td>
<td>1.42</td>
<td>1.45</td>
<td>1.32</td>
<td>2.41</td>
<td>1.54</td>
<td>1.85</td>
<td>1.72</td>
<td>1.51</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2.64</td>
<td>1.25</td>
<td>1.49</td>
<td>1.47</td>
<td>2.13</td>
<td>1.08</td>
<td>2.15</td>
<td>1.95</td>
<td>1.87</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3.07</td>
<td>1.84</td>
<td>1.16</td>
<td>2.09</td>
<td>1.88</td>
<td>0.99</td>
<td>2.14</td>
<td>1.12</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2.80</td>
<td>2.03</td>
<td>1.32</td>
<td>1.43</td>
<td>1.69</td>
<td>1.05</td>
<td>1.41</td>
<td>1.85</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
<td>16.45</td>
<td>17.85</td>
<td>15.44</td>
<td>17.61</td>
<td>22.58</td>
<td>14.33</td>
<td>17.84</td>
<td>18.26</td>
<td>18.59</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>1.65</td>
<td>1.79</td>
<td>1.54</td>
<td>1.76</td>
<td>2.26</td>
<td>1.43</td>
<td>1.78</td>
<td>1.83</td>
<td>1.86</td>
</tr>
</tbody>
</table>
6.5 Comparison with other Related Metrics

Having determined the optimized value of alpha to be 0.125 for this scenario (as shown in the table and graph in section 6.4), further test were carried out to validate the network performances of the proposed RSSI-aware metric. Throughput and delay are both key network performance parameters and these were tested on rETT and compared with hop counts and ETX.

6.5.1 Throughput Test

OLSR version 0.4.9 is capable of running routing metric using hop counts and as well as ETX metric. A clean version was obtained online (open source) and then compiled, the compiled version was then distributed to the mesh nodes already running the rETT metric. The RSSI-aware metric was then disabled on the mesh nodes while the original 0.4.9 version was installed and configured accordingly. Throughput and delay tests were then carried out under similar link conditions. After this rETT was in turn reinstated on the nodes and similar tests were repeated under similar link conditions.

This experiment was setup under the afternoon environmental condition and the network setup was similar to the performance optimisation test. The measuring tool was jperf and this was used to measure the end-to-end throughput from node 1 to node 5. rETT with $\alpha = 0.125$ was used for the RSSI-aware metric. OLSR version 0.4.9 with LinkQualityLevel value on the olsr.conf set to 2 was used to achieve the ETX metric. To achieve the hop count metric, the same version of OLSR with LinkQualityLevel set to 0 was compiled. These metrics were activated on the nodes and the throughput measurements were taken in turn. This process was repeated 5 times and the throughput for each of the metric and at each of the rounds were recorded.

Table 6.2 and Figure 6.20 show the results of the throughput test. Hop counts returned the least throughput with average of 0.59 Mbps. ETX came next with an average of 1.07 Mbps while rETT returned the best of the three with an
average of 1.30 Mbps. With *Hop counts*, all nodes are neighbours irrespective of the link quality provided the link is available. ETX considered the lost probability in the estimation of its link quality but this are only based on HELLO packets. *rETT* estimated its metric not only on lost probability but also considered the link capacity through RSSI measurement to estimate the link quality metric.

The results estimated that *rETT* improved the average network throughput of the network by more than double that of hop counts (1.30 Mbps and 0.59 Mbps respectively). Also, an improvement of 21.31% was recorded when compared with link metric ETX (1.30 Mbps and 1.07 Mbps respectively).

**Table 6.2:** Throughput comparison between the metrics

<table>
<thead>
<tr>
<th>Runs</th>
<th>Hop Counts</th>
<th>ETX</th>
<th>rETT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.54</td>
<td>0.93</td>
<td>1.31</td>
</tr>
<tr>
<td>2</td>
<td>0.65</td>
<td>1.31</td>
<td>1.25</td>
</tr>
<tr>
<td>3</td>
<td>0.70</td>
<td>1.13</td>
<td>1.24</td>
</tr>
<tr>
<td>4</td>
<td>0.51</td>
<td>1.01</td>
<td>1.38</td>
</tr>
<tr>
<td>5</td>
<td>0.55</td>
<td>0.97</td>
<td>1.31</td>
</tr>
<tr>
<td>Total</td>
<td><strong>2.95</strong></td>
<td><strong>5.35</strong></td>
<td><strong>6.49</strong></td>
</tr>
<tr>
<td>Average</td>
<td>0.59</td>
<td>1.07</td>
<td>1.30</td>
</tr>
</tbody>
</table>
6.5 Comparison with other Related Metrics

![Average Throughput Comparison](image)

**Figure 6.20**: Bar chart comparing mean throughputs for different metrics

### 6.5.2 Delay Test

A similar procedure was followed in terms of the implementation of the OLSR metrics. For the delay tests, ping statistics were used. This includes the total transmission time for the 60 packets. Also, the minimum, average, maximum and standard deviation of the statistics were obtained. All these parameters can be obtained through the ping command.

Ping statistics are measured in milliseconds. Each ping leaves the sending node every second and it measures the time it takes the ping packet to travel to the destination and back to the sending node. If the destination node is not available the ping reports “destination unreachable” or “timed out”. When the node is active, the ping reports the state of the link by the round trip time, also known as echo time. The average echo time is used in networks to estimate the delay in any network. It is important for networks to have good response time, else packets may be subject to retransmission or some packets may be lost as a result.

In this experiment, the same network topology as the throughput test and the
afternoon environment was setup. The 60 ping packets were sent from node 1 to node 5 with the different metrics running on the nodes. This was done in turn with hop-counts, ETX and rETT metrics. The topology or the path for reaching node 5 from node 1 depends on the estimation of the routing metric and how this resulted in the construction of the routing table. Hop count will attempt to send the packets directly to node 5 because the node can be reached by HELLO packets from node 1 due to the distance between them.

Table 6.3 and Figure 6.21 show the results of the ping response to estimate the network delay for the mesh node. This also compares the proposed rETT metric with the classical routing metric by hop counts and the link quality metric by ETX.

Table 6.3: Comparison of the delay for the different metrics

<table>
<thead>
<tr>
<th>Ping response time (ms)</th>
<th>Hop Counts</th>
<th>ETX</th>
<th>rETT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time</td>
<td>58999</td>
<td>59001</td>
<td>59004</td>
</tr>
<tr>
<td>Min</td>
<td>2.094</td>
<td>1.952</td>
<td>1.749</td>
</tr>
<tr>
<td>Avg</td>
<td>73.940</td>
<td>69.174</td>
<td>49.614</td>
</tr>
<tr>
<td>Max</td>
<td>607.834</td>
<td>436.391</td>
<td>632.451</td>
</tr>
<tr>
<td>Std</td>
<td>103.954</td>
<td>99.066</td>
<td>89.203</td>
</tr>
</tbody>
</table>

It can be observed that the total time for the three metrics are approximately the same (≈ 60seconds). This is because the ping packets are released from the sending node every second irrespective of the routing metrics. Considering the average delay for the network under different metrics, hop counts has the highest average delay with 73.94 milliseconds, followed closely by ETX with 69.17 milliseconds and the least average delay was achieved with rETT with 49.61 milliseconds. This translates to 33% improvement in network delay with hop counts and 28% also in delay compared with ETX link quality metrics. It can also be observed that the standard deviation was higher in hop counts.
than the other two metrics. This is because hop count is not a link quality metric and consistent attempts to send the packets directly to the destination can result in further delays and retransmissions. rETT has a lower standard deviation and it means that link was able to be consistent in finding a quality path to the destination and the deviation from the average delay is lower.

![Average Delay Comparison](image)

**Figure 6.21:** Bar chart comparing the delay measurement between the metrics

### 6.6 Summary

This chapter combined all the studies in the previous chapters as it involved the implementation of the proposed metric for the routing protocol for WMN. The determination and implementation of the EWMA filter for the RSSI function is discussed in this chapter. The proposed rETT metric works well with OLSR routing protocol and has shown efficient performance in terms of topology formation when compared with hop counts and ETX routing metrics.

More importantly, the optimisation of the routing metric was determined. This was carried out by empirical method and the throughput response at alpha value of 0.125 was observed to have returned the performance in this
setup. This value of alpha will be adopted in the final version of the routing protocol. The optimised $rETT$ metric was then compared with the classical routing metric by hop counts and link quality metric by ETX. Results show that $rETT$ throughput performance is more than double (120%) that of hop counts and 21.31% better than that of ETX metric. Also the result of the delay test estimated an improvement of 33% when compared with hop counts and 28% when compared with ETX metric.
Chapter 7

Contributions & Future Works

This presents the research contributions of the study and some areas identified with potential for further work. Some conclusions and recommendations are also discussed here.

7.1 Research Contributions

The aim and the specific objectives of this research stated in section 1.3 are revisited here to justify the research contributions from this study.

Objective 1: To study routing in WMN and analyse the shortcomings in the classical and present solutions to justify the need for a more efficient solution and approach.

The background and literature review of routing in WMN was highlighted in chapter 2. Link Quality routing methodologies through theoretical (Liu et al., 2007; Wu et al., 2009), simulation (Shang et al., 2009; Song et al., 2009) and by implementation (Carrera et al., 2009; Draves et al., 2004b) were identified. It was observed that some of the theoretical approaches have some constraints in modelling the channel condition which could not be incorporated into the architecture or practically implemented. Likewise, simulation tools most of...
the time failed to give realistic and accurate results due to imperfect channel modelling used by the tool (Chen et al., 2007). Hence routing solutions through implementation gives the most realistic results.

Further review related to the proposed link quality metric was critically set up chronologically to track the research developments in this area link quality routing up to date. The reviewed metrics are ETX (De Couto et al., 2003b), RTT (Adya et al., 2004), ETT & WCETT (Draves et al., 2004b), MIC (Yang et al., 2005a,b), mETX (Koksal and Balakrishnan, 2006), iAWARE (Subramanian et al., 2006), EETT (Jiang et al., 2007), ALM (Wang and Lim, 2008) and NBLC (Liu and Liao, 2008). The literature review brought to focus the merits and demerits of related studies. It also highlights areas of further works. It was observed that no work has a perfectly completed solution; there are advantages and disadvantages in each of them. This brought to focus and updated the research activities and development in the area of link-quality routing for WMN. It can be of significant importance and guidance to new researchers interested in optimising routing in WMN as it provides the necessary background for further studies.

**Objective 2: To propose an enhanced link quality routing metric based on cross-layer design that will involve the received signal strength information in the routing process for WMN.**

The design and methodology of the proposed routing metric was discussed in chapter 3 while routing by proactive and reactive approaches were highlighted in chapter 2. The basics of RSSI technology was also discussed in chapter 2. The proposed metrics were based on plug-in approach to extend the routing process of OLSR. The plug-in program was designed to perform the following tasks:

1. *Extracting the MAC Address and the RSSI of the neighbouring stations.*

2. *Updating the station lists and their RSSI values, then store the old and the new values.*
7.1 Research Contributions

3. Filtering the RSSI value by an exponential moving average filter.
4. Converting the average RSSI value to dBm range.
5. Transforming the RSSI value to bit-rate based on the received sensitivity of the chipset.

It should be noted that the “real” bit-rate which can also be extracted from the Wi-Fi chipset is not used in this design. This is because these values change at a rather faster rate and it will make the routing process unstable. The ability to control the filtering level of the RSSI extracted from the chipset and then transform them to rate value based on the sensitivity limits is a more scalable approach.

The plug-in was implemented into the process of OLSR to enhance the metric calculation and topology formation in order to improve its performance. The procedure or the overall methodology for the proposed metric implementation followed the following procedure.

1. Identification of a metric for the OLSR protocol that involves the RSSI, which is proportional to the bit-rate due to the sensitivity limits.
2. Retrieval of the RSSI information from the wireless card driver through IOCTL calls.
3. Analysing the variability of the measured RSSI of the mesh nodes in the network.
4. Implementation of the RSSI-based metric in the OLSR, using the signal strength information received through the IOCTL calls.
5. Optimizing the routing performance by determining the filtered level of the RSSI

Objective 3: To study the characteristics of the RSSI samples in order to clarify and justify its suitability for the proposed approach.
7.1 Research Contributions

Several experiments were conducted to study the characteristics of RSSI samples and they are documented in chapter 5. The statistical analyses of the links at different conditions were studied. These were evaluated by the use of sample histograms, auto-correlations of the samples and the probability density functions of the measurement data. Firstly, it was confirmed that RSSI samples are not random samples but signals with some variability.

The link conditions are translated in the measured samples and the behaviours of the statistical data are obviously in agreement with link conditions under different scenarios. The histogram of the sample also suggests that high RSSI values indicate a good signal while low RSSI values indicate a low signal. The effect on interference by introduction of noise into the channel and the effect of different transmit power levels on the received signal strength was evaluated. The signal strength between the nodes was increased by the increment in the power level. However, the effects of noise and increment in the transmit power on the received signal did not show a strong correlation when the link variability was investigated. Although increments in the transmit power increased the signal strength slightly for both interference and without interference conditions.

**Objective 4:** To incorporates the proposed solution into optimised link state routing protocol (OLSR) to extend and enhance its routing process.

The implementation of the RSSI-aware metric on OLSR was documented is chapter six. The testing and determination of suitable RSSI filter was implemented. Furthermore, this was incorporated into OLSR process and hence the metric for the calculation of link quality was driven by the processed RSSI embedded into the routing process.

The implementation was in form of expected transmission time, i.e. the metric is a function of time and is measured in seconds. The RSSI-to-bit-rate information on the routing metric is used in the computation of the routing table. The routing process aims at choosing the link with the lowest $rETT$ value. For an end-to-end transmission which will involve several links, the path with the least path metric $rETT_p$ is chosen by the routing algorithm and hence used...
in the formation of topology for the WMN. Depending on the link condition, \( rETT \) updates the routing metric and this changes the topology accordingly. Hence, the routing protocol has the required information to choose the better link for packet transmission.

**Objective 5:** To implement the RSSI-aware metric on implementation test-beds in order to obtain realistic routing performance for the proposed solution.

The literature review highlights the important of obtaining realistic results for routing solution (Carrera et al., 2009; Chen et al., 2007; Iannone et al., 2004). Hence, the details of the experimental setup and the implementation test-bed for this investigation were described in chapter four. Routing is a practical networking process and hence the need to obtain realistic results motivated the test-bed approach. This was implemented on Linux boxes running Ubuntu and \( olsrd \). The wireless driver used the MadWifi 0.9.4 driver which made it possible to perform some kernel level operation which aided this objective. All software used in this research is open source. The environmental conditions for the experiments were described in chapter 4. The effects of multi-path and human traffic movement in the laboratory were evident in the results obtained in chapters 5 and 6.

The experimentation results in chapter 5 and the implementation results in chapter 6 justify the practicability of this solution. The implementation approach also helps in studying the dynamics of the RSSI filter which was introduced in subsection 3.1.3 and implemented in section 6.1. More importantly, during the routing implementation experiments, the test-beds responded well to the changes in link conditions and this gave much credibility to this investigation. It can be recommended that theoretical and simulation solutions should endeavour to validate results through implementation to justify their results.

**Objective 6:** To evaluate and optimise the performances of the proposed RSSI-aware metric and compare with the classical and related link quality routing solutions.
7.2 Transferability

The performances of the proposed routing metric called \( rETT \) was evaluated and compared with classical routing approach by hop count and also the link quality approach by ETX. The RSSI-aware metric outperformed both of them in choosing the better link for the routing path on the WMN.

The performance of the RSSI-aware metric was also optimised by varying the filter constants and the testing the performance of the network by carrying out throughput response tests. The optimisation by empirical method resulted in choosing alpha constant 0.125 as the best filter constant that produced the best network performance under the investigated implementation test-bed and environment. This is documented in section 6.4.

The network throughput and delay performances of three metrics were also compared. \( rETT \) improved the network throughput by 120% when compared with the classical hop counts metric and an improvement of 21% when compared with ETX link quality metric. In terms of delay, the improvement on hop counts is 33% while \( rETT \) improved the delay by 28% when compared with ETX metric. This evaluation was presented in section 6.5.

7.2 Transferability

The scope for transferring the research contributions of this research into other areas or application is discussed in this section. This is discussed in the reverse order of the last section.

Routing in MANet and WMN is quite practical in nature and this research work has proposed and tested an enhanced routing metric to extend the efficiency of a routing protocol. This work could be registered as RFC in the near future but meanwhile work will continue on the enhancement of the metric. This enhancement will include deployment on a larger and wider WMN. OLSR is registered as RFC 3626, and since RFC will not accept amendments like this to be registered on the same RFC number, then this work can be registered as a new RFC.
The methodology used in this study is an implementation approach. This gave much credence to the obtained results. The details of the experimental setup were documented in chapter 4. This approach, though not entirely new, but the procedure and the good practice exhibited by this research can be a guideline to other researchers interested in developing this area in the future. This will encourage repeatability.

Extending and enhancing a routing protocol like OLSR is an interesting exercise. Other areas of network features like security, QoS etc. can be researched using this optimised routing protocol. The process of using the plug-in approach in achieving such a task is a tested procedure which this work has exhibited. This can be leveraged and adapted into other area of research depending on the goal or set objective of the research.

Results obtained when studying the characteristics of RSSI sample is another success of this work. These show the potentials of RSSI and were able to reveal some of the nature of this parameter. These includes the effect of multi-path and human traffic on the received signal strength and the randomness of the sample. How these signals behave under different link conditions is not only applicable or transferable to routing techniques but can also be used in designing other aspects of wireless or mobile networking. Similar behaviour of RSSI exists in sensor networks and the results obtained can guide other future research goals. Furthermore, these results were obtained from implementation test-beds and they are practicable and applicable.

The design of a novel routing metric based on RSSI is a challenging exercise. Due to the variability of RSSI, past researches in this area have tended to avoid this direction (Vlavianos et al., 2008). This research has revealed some benefits of RSSI and in this research, it worked efficiently with loss probability to estimate metric for routing in WMN with improved performance. This is exhibited in this research and may usher in a new technique for further studies in WMN technologies.
7.3 Conclusion

In conclusion, this work has investigated the shortcomings of recent link-quality routing process applicable to WMN and has identified areas of improvements. The research has proposed, design and implemented a novel metric approach to enhance the OLSR routing protocol. This was implemented on test-beds and performances were evaluated. It showed that the proposed RSSI-aware metric performed better than fundamental routing technique by hop counts and the link quality approach by ETX. This work was adequately optimised in order to achieve the best network performance through this process. The characteristics of RSSI were analysed, clarified and classified in this research and the true nature of this parameter was evaluated through the practical process employed by this study.

The contributions through this research are summarised here:

- A review of major contributions to routing solutions by theoretical (Liu et al., 2007; Wu et al., 2009), simulation (Shang et al., 2009; Song et al., 2009) and implementation approaches (Carrera et al., 2009; Draves et al., 2004b). Further review of cross-layer routing approaches highlighting the merits and demerits of each contributions and also highlighting the gaps in the following solutions ETX (De Couto et al., 2003b), RTT (Adya et al., 2004), ETT & WCETT (Draves et al., 2004b), MIC (Yang et al., 2005a,b), mETX (Koksal and Balakrishnan, 2006), iAWARE (Subramaniam et al., 2006), ETT (Jiang et al., 2007), ALM (Wang and Lim, 2008) and NBLC (Liu and Liao, 2008).

- An enhanced RSSI-aware link quality routing metric was proposed. The design of this metric involves extraction of RSSI from the neighbouring stations, filtering the samples by EWMA filter, and transforming the averaged value to bit-rate based on sensitivity limits of the chipset (see subsection 3.1.5. The metric was optimised by determining the optimum filter level for the RSSI signal.
• A study of the characteristic of RSSI was conducted. Statistical analysis of RSSI sample reveals the response of RSSI to link conditions. The variances of the samples also suggest the effect of multi-path on the link conditions. The effect of noise and transmit power level on RSSI signal was also evaluated.

• The proposed solution was implemented on OLSR. The \( rETT \) metric which is a function RSSI drives the metric of the routing protocol (see 6.3. The evaluation of the metric captures the link conditions. Hence, the metric adapts the routing process in line with the link condition for efficient routing operation.

• The implementation of the proposed solution on test-beds established the practicability of the results. Experimentation and implementation results show the realistic response of the routing process to link changes under different scenarios.

• Experimental results show the improvement of the \( rETT \) metric over the classical hop counts metric and the ETX link quality metric. This was in terms of topology formation, network throughput and delay.

### 7.4 Further Works

This work can be extended in many directions. As it has been observed in this work, the implementation of routing protocol on test-beds is quite challenging and time-consuming. Therefore, the practicability of any extension needs to be carefully addressed. Although, further theoretical study on this solution is needed, this research would be an important guidance for further works.

Further attempts can be made to develop an adaptive algorithm based also on the weighted RSSI, e.g., a Kalman filter (Simon, 2010) that takes into account the variability of the channel.
It should be noted that the variability of the channel needs to be controlled as it triggers re-computations of the network topology every time the Kalmar filter signals a change. The implementation of the algorithm needs to take into consideration the stability of the routing protocol. Adopting the Kalman filter directly will not make the routing converge and this will hamper the network performance because the topology re-computation will consume network resources. This is a challenging task but it is worth considering as a further work.

Another area of interest is to study the impact of the incorporation of antenna diversity at all nodes in the network. This can be achieved by using two antennas (double radios) at each node to improve the reliability and quality of the wireless link. With the use of two radios, each antenna can transmit and receive on a different frequency and hence avoid packet collision. Also, this research shows the effect of multi-path on the received signals since all nodes are not always within line-of-sight. Consequently, signals are reflected along multiple paths before getting to their respective destinations. This introduces phase shifts, attenuation and time delays. Antenna diversity or space diversity can address some of this problem and hence improve network performance. It should be noted that this will introduce additional cost to the network setup. The net effect of the gain through this process should be compared with the additional cost of implementing this solution to justify it.

Another option is to consider the implementation of the rETT metric on IEEE 802.11n. This is an amendment to the IEEE 802.11 technology, adding multiple-input multiple-output antenna (MIMO).

This metric working as plug-in was implemented on OLSR version 0.4.9 which was a stable version at the time the experiments were conducted. There are more recent versions which are now stable and their functionality can be benefited in the overall performance of the network. Hence, the upgrade of the plug-in to work with more recent versions of the routing protocol is currently being considered.
Further theoretical approaches can also be investigated. Devising an analytical model for the alpha coefficient for different theoretical probability distributions of RSSI and ETX could be considered. Some of the results obtained on the characteristics of the links by RSSI samples and further experiments to investigate the ETX estimations for the links under different conditions can be used to build this model. This model can be optimised theoretically and hence obtain the optimum alpha or filter constant to yield the best network performance.
References

radio unification protocol for IEEE 802.11 wireless networks, in ‘Broadband

Aguayo, D., Bicket, J. and Morris, R. (2005), ScrRR: A high throughput rout-
ing protocol for 802.11 mesh networks (DRAFT), Technical report, MIT.


Aldalgamouni, T. and Elhakeem, A. (2009), A joint cross-layer routing and re-
on’, pp. 349–354.

Anzai, D. and Hara, S. (2009), An RSSI-Based MAP Localization Method with
Channel Parameters Estimation in Wireless Sensor Networks, in ‘Vehicular

Bardwell, J. (2004), The truth about 802.11 signal and noise metrics, Technical
report.

URL: http://madwifi-project.org/attachment/wiki/UserDocs/
RSSI/you_believe_D100201.pdf

Basagni, S., Chlamtac, I., Syrotiuk, V. R. and Woodward, B. A. (1998), A
Distance Routing Effect Algorithm for Mobility (DREAM), in ‘Proceedings
REFERENCES


URL: http://doi.acm.org/10.1145/288235.288254


Carleton University WMN (2011), ‘Wireless Mesh Network at Carleton University’.

URL: http://kunz-pc.sce.carleton.ca/MESH/index.htm


URL: http://doi.acm.org/10.1145/1298126.1298155

163

**URL:** [http://www.ietf.org/rfc/rfc3626.txt](http://www.ietf.org/rfc/rfc3626.txt)


**URL:** [http://doi.acm.org/10.1145/938985.939000](http://doi.acm.org/10.1145/938985.939000)


**URL:** [http://doi.acm.org/10.1145/1023720.1023732](http://doi.acm.org/10.1145/1023720.1023732)


REFERENCES

URL: http://www.ece.gatech.edu/research/labs/bwm/mesh/testbed.html


IEEE P802.11s/D1.08 (2008), Draft STANDARD for Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Technical report.

URL: http://spectrum.ieee.org/podcast/telecom/internet/long-range-wifi-network-aids-haiti-relief-effort

URL: http://www.isi.edu/nsnam/ns/

URL: http://iperf.sourceforge.net/


URL: http://www.ietf.org/rfc/rfc4728.txt


URL: http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8687


URL: http://pdos.csail.mit.edu/roofnet/design/

Ni, X. (2008), Performance evaluation of ETX on grid based wireless mesh networks (MPhil Thesis Report), PhD thesis, University of New South Wales, Australia, Electrical Engineering & Telecommunications, Faculty of Engineering UNSW.


URL: http://www.ietf.org/rfc/rfc3561.txt


URL: http://doi.acm.org/10.1145/190314.190336


URL: http://www.scalable-networks.com/


Appendix A

WLAN Cards Specifications

A.1 TP-Link TL-WN650G

108Mbps Super G & eXtended Range Wireless PCI Adapter

Software Specification

Standards:
IEEE 802.11g, IEEE 802.11b

Wireless Signal Rates With Automatic Fallback:
11g: Up to 108Mbps (dynamic)
11b: Up to 11Mbps (dynamic)

Frequency Range:
2.4-2.4835GHz

Wireless Transmit Power:
20dBm (MAX)

Modulation Type:
1M DBPSK, 2M DQPSK, 5.5M/11M CCK,
6M/9M/12M/18M/24M/36M/48M/54M OFDM

Receiver Sensitivity:
108M: -68dBm@10% PER
54M: -68dBm@10% PER
11M: -85dBm@8% PER  
6M: -88dBm@10% PER  
1M: -90dBm@8% PER  
256K: -105dBm@8% PER  

Work Mode:  
Ad-Hoc  
Infrastructure  

Wireless Range Indoors:  
up to 200m,  
Outdoors up to 830m.  

Wireless Security:  
64/128/152 bit WEP  
WPA/WPA2, WPA-PSK/WPA2-PSK (TKIP/AES)  

Support Operating System:  
Windows 98SE/ME/2000/XP/Vista  

**Hardware Specification**  

Interface:  
32-bit PCI connector  

Antenna Type:  
2dBi Fixed Omni-directional Antenna  

Certifications:  
CE, FCC  

Operating Temperature:  
0 °C~40 °C (32°~104°)  

Storage Temperature:  
-40 °C~70 °C (-40°~158°)  

Relative Humidity:  
10% ~ 90%, non condensation  

Storage Humidity:  
5%~95% non-condensing  

Dimensions:  
5.2 × 4.8 × 0.9 in. (133 × 121 × 22 mm)
## A.2 DWL-G650

High Speed 2.4GHz (802.11g) Wireless 108Mbps Cardbus Adapter

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Standards</th>
<th>Bus Type</th>
<th>Wireless Signal Rates*</th>
<th>Security</th>
<th>Media Access Control</th>
<th>Frequency Range</th>
<th>Wireless Signal Range*</th>
<th>Power Consumption</th>
<th>Modulation Technology</th>
<th>Receiver Sensitivity*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IEEE 802.11</td>
<td>32-bit Cardbus</td>
<td>D-Link 108G: 108Mbps</td>
<td>64-, 128-WEP</td>
<td>CSMA/CA with ACK</td>
<td>2.4 GHz to 2.462 GHz</td>
<td>Indoors: Up to 328 feet (100 meters)</td>
<td>Standby mode = 4.66 mA</td>
<td>Orthogonal Frequency Division Multiplexing (OFDM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IEEE 802.11b</td>
<td></td>
<td></td>
<td></td>
<td>WPA PSK (Pre-Shared Key)</td>
<td></td>
<td>Outdoors: Up to 1,312 feet (400 meters)</td>
<td>Transmit mode = 248 mA</td>
<td>54 Mbps OFDM, 10% PER, -68 dBm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IEEE 802.11g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PowerSave mode = 28 mA</td>
<td>48 Mbps OFDM, 10% PER, -68 dBm</td>
<td>48 Mbps OFDM, 10% PER, -68 dBm</td>
<td></td>
</tr>
<tr>
<td><strong>Bus Type</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>36 Mbps OFDM, 10% PER, -75 dBm</td>
<td>36 Mbps OFDM, 10% PER, -75 dBm</td>
<td></td>
</tr>
<tr>
<td>*<em>Wireless Signal Rates</em></td>
<td>D-Link 108G: 108Mbps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24 Mbps OFDM, 10% PER, -79 dBm</td>
<td>24 Mbps OFDM, 10% PER, -79 dBm</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18 Mbps OFDM, 11 Mbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12 Mbps OFDM, 9 Mbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11 Mbps OFDM, 6 Mbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9 Mbps OFDM, 5.5 Mbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 Mbps OFDM, 2 Mbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.5 Mbps OFDM, 1 Mbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 Mbps OFDM, 64-, 128-WEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Mbps OFDM, 802.1x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireless Transmitter Power</td>
<td>15 dBm 2dB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Antenna Type</td>
<td>Dual Antenna Diversity Switching</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>32F to 131F (0C to 55C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>95% maximum (non-condensing)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dimensions</td>
<td>L = 4.64 inches (114.3 mm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W = 2.13 inches (54 mm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H = 0.34 inches (8.7 mm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>0.12 lbs (55 g)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certifications</td>
<td>FCC part 15b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warranty</td>
<td>3 Year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Maximum wireless signal rate derived from IEEE Standard 802.11g specifications. Actual data throughput will vary. Network conditions and environmental factors, including volume of network traffic, building materials and construction, and network overhead, lower actual data throughput rate. Environmental factors will adversely affect wireless signal range.

### A.3 Wistron Neweb CM9 Atheros

802.11a/b/g Dualband mPCI 5004chipset (CM9)

Frequency Range:
- USA: 2.400 - 2.483GHz, 5.15 ~ 5.35Ghz, 5.725 ~ 5.825Ghz
- Europe: 2.400 - 2.483GHz, 5.15~ 5.35Ghz, 5.47 ~ 5.725Ghz
- Japan: 2.400 - 2.483GHz, 4.90 - 5.091GHz, 5.15 - 5.25GHz
- China: 2.400 - 2.483GHz, 5.725 ~5.85Ghz

Modulation Technique:
802.11b/g
- DSSS (DBPSK, DQPSK, CCK)
- OFDM (BPSK, QPSK, 16-QAM, 64-QAM)

802.11a
- OFDM (BPSK, QPSK, 16-QAM, 64-QAM)

Host Interface:
- Mini-PCI form factor; Mini-PCI Version 1.0 type 3B

Channels Support:
802.11b/g
- US/Canada: 11 (1 ~ 11)
- Major European country: 13 (1 ~ 13)
- France: 4 (10 ~ 13)
- Japan: 11b: 14 (1~13 or 14th), 11g: 13 (1 ~ 13)
- China: 13 (1 ~ 13)

802.11a
- US/Canada: 12 non-overlapping channels (5.15 ~ 5.35GHz, 5.725 ~ 5.825GHz)
- Europe: 19 non-overlapping channel (5.15 ~ 5.35GHz, 5.47 ~ 5.725GHz)
- Japan: 4 non-overlapping channels (5.15 ~ 5.25GHz)
- China: 5 non-overlapping channels (5.725 ~ 5.85GHz)

Operation distance (depend on antenna performance):
802.11a
Outdoor: 85m@54Mbps, 300m@6Mbps
Indoor: 20m@54Mbps, 40m@6Mbps

802.11b
Outdoor: 300m@11Mbps, 400m@1Mbps
Indoor: 30m@11Mbps, 50m@1Mbps

802.11g
Outdoor: 80m@54Mbps, 300m@6Mbps
Indoor: 15m@54Mbps, 35m@6Mbps

Receive Sensitivity:
802.11a
-88dBm@6Mbps, -87dBm@9Mbps, -85dBm@12Mbps, -83dBm@18Mbps, -80dBm@24Mbps,
-75dBm@36Mbps, -73dBm@48Mbps, -71dBm@54Mbps

802.11b
-95dBm@1Mbps, -94dBm@2Mbps, -92dBm@5.5Mbps, -90dBm@11Mbps

802.11g
A.3 Wistron Neweb CM9 Atheros

-90dBm@6Mbps, -89dBm@9Mbps, -87dBm@12Mbps, -85dBm@18Mbps, -82dBm@24Mbps, -79dBm@36Mbps, -76dBm@48Mbps, -74dBm@54Mbps

Security:
- 64-bit, 128-bit, 152-bit WEP Encryption
- 802.1x Authentication
- AES-CCM & TKIP Encryption

Operation Mode:
Infrastructure & Ad-hoc mode

Transfer Data Rate:
802.11b/g: 11, 5.5, 2, 1 Mbps, auto-fallback, up to 54 Mbps
802.11g (Super mode): up to 108 Mbps
802.11a (Normal mode): 54, 48, 36, 24, 18, 12, 9, 6Mbps, auto-fallback
802.11a (Turbo mode): 108, 96, 72, 48, 36, 24, 18, 12 Mbps, auto-fallback

Operation Temperature:
0 ~ 70

Storage Temperature:
-20 ~ 80

Wi-Fi Alliance:
WECA Compliant

WHQL:
Microsoft XP Compliant

FAA:
S/W audio On/Off support

EMC Certificate:
FCC part 15 (USA), with multiple e-Antenna
Telec (Japan), with multiple e-Antenna

Media Access Protocol:
CSMA/CA with ACK architecture 32-bit MAC

Antenna connector:
2 x UFL Ultra-miniature coaxial connectors
Appendix B

Typical OLSR Configuration file

#
# olsr.org OLSR daemon config file
#
# Lines starting with a # are discarded
#
# This file was shipped with olsrd 0.4.9
#
# This file is an example of a typical
# configuration for a mostly static
# network(regarding mobility) using
# the LQ extention
#
# Debug level(0-9)
# If set to 0 the daemon runs in the background
DebugLevel 2
#
# IP version to use (4 or 6)
IpVersion 4
#
# Clear the screen each time the internal state changes
ClearScreen yes
#
# HNA IPv4 routes
# syntax: netaddr netmask
# Example Internet gateway:
# 0.0.0.0 0.0.0.0

Hna4
{
# Internet gateway:
# 0.0.0.0 0.0.0.0
# more entries can be added:
# 192.168.1.0 255.255.255.0
}

# HNA IPv6 routes
# syntax: netaddr prefix
# Example Internet gateway:
Hna6
{
# Internet gateway:
# :: 0
# more entries can be added:
# fec0:2200:106::: 48
}

# Should olsrd keep on running even if there are
# no interfaces available? This is a good idea
# for a PCMCIA/USB hotswap environment.
# "yes" OR "no"
AllowNoInt yes

# TOS(type of service) value for
# the IP header of control traffic.
# If not set it will default to 16

#TosValue 16

# The fixed willingness to use(0-7)
# If not set willingness will be calculated
# dynamically based on battery/power status
# if such information is available

#Willingness 4

# Allow processes like the GUI front-end
# to connect to the daemon.
IpcConnect
Typical OLSR Configuration file

{
  # Determines how many simultaneously
  # IPC connections that will be allowed
  # Setting this to 0 disables IPC
  MaxConnections 0

  # By default only 127.0.0.1 is allowed
  # to connect. Here allowed hosts can
  # be added
  Host 127.0.0.1
  #Host 10.0.0.5

  # You can also specify entire net-ranges
  # that are allowed to connect. Multiple
  # entries are allowed
  #Net 192.168.1.0 255.255.255.0

  # Whether to use hysteresis or not
  # Hysteresis adds more robustness to the
  # link sensing but delays neighbour registration.
  # Used by default. ‘yes’ or ‘no’
  UseHysteresis no

  # Hysteresis parameters
  # Do not alter these unless you know
  # what you are doing!
  # Set to auto by default. Allowed
  # values are floating point values
  # in the interval 0,1
  # THR_LOW must always be lower than
  # THR_HIGH.
  #HystScaling 0.50
  #HystThrHigh 0.80
  #HystThrLow 0.30

  # Link quality level
  # 0 = do not use link quality
  # 1 = use link quality for MPR selection
  # 2 = use link quality for MPR selection and routing
}
# Defaults to 0
LinkQualityLevel 2

# Link quality window size
# Defaults to 10
LinkQualityWinSize 10

# Polling rate in seconds(float).
# Default value 0.05 sec
Pollrate 0.05

# TC redundancy
# Specifies how much neighbour info should
# be sent in TC messages
# Possible values are:
# 0 - only send MPR selectors
# 1 - send MPR selectors and MPRs
# 2 - send all neighbours
#
# defaults to 0
TcRedundancy 2

# MPR coverage
# Specifies how many MPRs a node should
# try select to reach every 2 hop neighbour
#
# Can be set to any integer >0
#
# defaults to 1
MprCoverage 3

# Olsrd plug-ins to load
# This must be the absolute path to the file
# or the loader will use the following scheme:
# - Try the paths in the LD_LIBRARY_PATH
# environment variable.
# - The list of libraries cached in /etc/ld.so.cache
# - /lib, followed by /usr/lib
# Example plug-in entry with parameters:

#LoadPlug-in "olsrd.dyn_gw.so.0.3"
#
# Here parameters are set to be sent to the
# plug-in. These are on the form "key" "value".
# Parameters of course, differs from plug-in to plug-in.
# Consult the documentation of your plug-in for details.
#
# Example: dyn_gw params
#
# how often to check for Internet connectivity
# defaults to 5 secs
# PlParam "Interval" "40"
#
# if one or more IPv4 addresses are given, do a ping on these in
# descending order to validate that there is not only an entry in
# routing table, but also a real internet connection. If any of
# these addresses could be pinged successfully, the test was
# successful, i.e. if the ping on the 1st address was successful, the
# 2nd won’t be pinged
# PlParam "Ping" "141.1.1.1"
# PlParam "Ping" "194.25.2.129"
#
#
# Interfaces and their rules
# Omitted options will be set to the
# default values. Multiple interfaces
# can be specified in the same block
# and multiple blocks can be set.
#
# !!CHANGE THE INTERFACE LABEL(s) TO MATCH YOUR INTERFACE(s)!!
# (eg. wlan0 or eth1):

Interface "XXX" "YYY"
{
#
# IPv4 broadcast address to use. The
# one useful example would be 255.255.255.255
# If not defined the broadcast address
# every card is configured with is used
#
# Ip4Broadcast 255.255.255.255
#
# IPv6 address scope to use.
Typical OLSR Configuration file

# Must be 'site-local' or 'global'
# Ip6AddrType site-local
# IPv6 multicast address to use when
# using site-local addresses.
# If not defined, ff05::15 is used
# Ip6MulticastSite ff05::11
# IPv6 multicast address to use when
# using global addresses
# If not defined, ff0e::1 is used
# Ip6MulticastGlobal ff0e::1

# Emission intervals,
# If not defined, RFC proposed values will
# be used in most cases.
# Hello interval in seconds(float)
HelloInterval 2.0
# HELLO validity time
HelloValidityTime 20.0
# TC interval in seconds(float)
TcInterval 5.0
# TC validity time
TcValidityTime 30.0
# MID interval in seconds(float)
MidInterval 5.0
# MID validity time
MidValidityTime 30.0
# HNA interval in seconds(float)
HnaInterval 5.0
# HNA validity time
HnaValidityTime 30.0

# When multiple links exist between hosts
# the weight of interface is used to determine
# the link to use. Normally the weight is
# automatically calculated by olsrd based
# on the characteristics of the interface,
# but here you can specify a fixed value.
# Olsrd will choose links with the lowest value.

# Weight 0

}
Appendix C

More RSSI Experimental Results

This appendix contains more of the measurement data reported in chapters five and six under the RSSI experimentation and the $rETT$ implementation. These include the RSSI measurements under the morning, afternoon and evening environmental conditions. The histograms superimposed with PDFs, autocorrelations and the statistical analysis of the distribution are also reported. More measurements under the artificial interference condition and the rest of RSSI filtering experimentations are documented in this appendix.
C.1 Comparison of statistical data for the different environmental measurements

Figure C.1: The comparison of measurement data showing the differences in signal levels at different times of the day - 2
C.1 Comparison of statistical data for the different environmental measurements

**Figure C.2:** The comparison of measurement data showing the PDF superimposed with the theoretical Gaussian PDF - 2

**Figure C.3:** The comparison of measurement data showing the autocorrelation functions of the data - 2
C.1 Comparison of statistical data for the different environmental measurements

Table C.1: Table comparing the mean, standard deviation and range values of measured RSSI data

<table>
<thead>
<tr>
<th>Time of the day</th>
<th>Mean (dBm)</th>
<th>Std Dev. (dBm)</th>
<th>Range (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morning</td>
<td>-57.87</td>
<td>2.605</td>
<td>14</td>
</tr>
<tr>
<td>Afternoon</td>
<td>-69.92</td>
<td>3.239</td>
<td>18</td>
</tr>
<tr>
<td>Night</td>
<td>-57.85</td>
<td>2.280</td>
<td>14</td>
</tr>
</tbody>
</table>

Figure C.4: Mean, standard deviation and range values of RSSI measurement data at different times of the day - 2
C.1 Comparison of statistical data for the different environmental measurements

**Figure C.5:** The comparison of measurement data showing the differences in signal levels at different times of the day - 3

**Figure C.6:** The comparison of measurement data showing the PDF superimposed with the theoretical Gaussian PDF - 3
C.1 Comparison of statistical data for the different environmental measurements

![Graphs showing correlation functions for different times of the day.](image)

**Figure C.7**: The comparison of measurement data showing the autocorrelation functions of the data - 3

<table>
<thead>
<tr>
<th>Time of the day</th>
<th>Mean (dBm)</th>
<th>Std Dev. (dBm)</th>
<th>Range (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morning</td>
<td>-54.14</td>
<td>2.343</td>
<td>14</td>
</tr>
<tr>
<td>Afternoon</td>
<td>-63.05</td>
<td>3.361</td>
<td>16</td>
</tr>
<tr>
<td>Night</td>
<td>-53.64</td>
<td>2.018</td>
<td>12</td>
</tr>
</tbody>
</table>

**Table C.2**: Table comparing the mean, standard deviation and range values of measured RSSI data - 3
C.1 Comparison of statistical data for the different environmental measurements

Figure C.8: Mean, standard deviation and range values of RSSI measurement data at different times of the day.
C.2 Noise and transmit power effects on the measured RSSI

Figure C.9: Graphs of measurement data of a node under artificial interference with different transmit power settings - 2
C.2 Noise and transmit power effects on the measured RSSI

Figure C.10: Histogram & PDF of measurement data of a node under artificial interference with different transmit power settings - 2
C.2 Noise and transmit power effects on the measured RSSI

Figure C.11: Mean, standard deviation and range values of RSSI measurement data at different transmit power levels under artificial interference - 2
C.2 Noise and transmit power effects on the measured RSSI

![Graphs showing the measured RSSI for different transmit power settings](image1)

**Figure C.12:** Graphs of measurement data of a node without artificial interference with different transmit power settings - 3

![Histogram and PDF of measurement data](image2)

**Figure C.13:** Histogram & PDF of measurement data of a node without artificial interference with different transmit power settings - 3
C.2 Noise and transmit power effects on the measured RSSI

Figure C.14: Mean, standard deviation and range values of RSSI measurement data at different transmit power levels without artificial interference - 3
Appendix D

OLSR Components

The concept of multi-hop wireless networks routing is discussed in chapter two. An overview of OLSR routing protocol is also highlighted there. In this appendix, the protocol structure of OLSR will be further examined. The concept of protocol extension through plug-in approach which is the methodology of this research will be discussed.

D.1 OLSR Routing Protocol Structure

It is worth noting that this routing protocol is an optimisation of the classical link state routing approach based on Dijkstra’s algorithm. However, it has a link quality capability which this work is being leveraged upon for the protocol extension. Only the aspect of the protocol that relates to this is discussed in this thesis and is implemented accordingly in chapter six of this thesis. Other generic routing information is left out.

D.2 Information repositories

OLSR is a derivative of link state routing algorithm and it maintains its states by variety of information kept on its databases or routing tables. The information repositories are updated by triggers from control messages based on the state of the link and the information stored are likewise used when these messages are generated. The following are the relevant information repositories that are necessary for core OLSR functionalities. These are part of the OLSR components that are relevant to this research. Their parameters are particularly
examined and some modified to achieve the objectives of this research.

D.2.1 Link set

This is specifically maintained to calculate the state of the link to the respective neighbours on the network. This operates on the interface-to-interface links and it is the only database that works on non-main-addresses basis.

D.2.2 Multiple Interface Association Information Base

This stores interface addresses of nodes and are particularly useful when nodes have more than one communication interfaces.

D.2.3 Neighbour Set

This records all registered one-hop neighbours and they are dynamically updated based on the information on the link set database. They also register asymmetric as well as asymmetry neighbours.

D.2.4 2-hop Neighbour Set

Based on the MPR functionalities, this database registers all nodes except the local or originating node that can be reached via a one-hop neighbour. This set may contain some nodes already registered in the neighbour set database.

D.2.5 MPR Set

This repository registers all MPR selected by the local node. This optimisation process is discussed briefly in chapter two.

D.2.6 MPR Selector Set

This contains all nodes that have identified and selected this node as MPR.
D.2.7 Topology Information Base

This is a database with information of all link-state information received from neighbouring nodes on the OLSR routing domain.

D.2.8 Duplicate Set

Information on recently processed and forwarded messages is stored on this dataset.

D.3 Neighbour discovery

The task of discovery neighbours in wireless networks is a fundamental one and this is required for routing protocol to run efficiently. Also, the states of the communication links or channels are necessary information for every participating node on the network as they are used in making routing decisions. This can be achieved by sending HELLO messages on a regular interval. A simple illustration is found in the figure below. Node A first sends an empty HELLO message to node B, this is received and node B registers node A as an asymmetric neighbour because there is no address relating to node B in the HELLO message. Node B then send node A an HELLO message declaring node A as its asymmetric neighbour. Node A receives this message and confirms its address in the message and therefore sends node B an HELLO message as symmetric neighbour. At this time, node A has the address of node B in its database and this information is contained in the asymmetric HELLO message sent to node B. Node B consequently registers node A as a symmetric neighbour upon reception of the HELLO message.

HELLO messages can also be used for other functionalities apart from discovery neighbours in a network domain. Operations like link-sensing, neighbour-sensing, two-hop neighbour-sensing and MPR selector sensing. HELLO messages are generated on every interfaces participating in the network domain. These messages transmit information about all known links and neighbours and also declare the types of the neighbours. This declaration also includes which of the neighbouring nodes the originating node has chosen as its MPRs. All registered links and neighbours are grouped by the link and neighbours type to optimise the usage of the bytes.
D.4 Control traffic

Internet Assigned Number Authority (IANA) assigned UDP port 698 for all OLSR control traffic. While using IPv4, the RFC specifies that control traffic can be broadcasted on the specified port without specifying any broadcast address. This is slightly different with IPv6, as no broadcast exists then the control traffic uses a multicast address.

D.4.1 OLSR Packet Format

OLSR packets are fundamental units of OLSR traffic and are generally comprised of OLSR packet header and a body. The fields in the OLSR packets are already highlighted in section 2.1.5.

D.4.2 OLSR message types

OLSR core functionality defines three message types and these are HELLO message, Topology Control TC message and MID message. Details of these message types are documented in RFC 3626. The generation and processing of these messages are useful in defining any OLSR core functionality. Other custom packets as required by the designer are permitted in OLSR and are forwarded according to the default forwarding rule.
D.5 Route calculation in OLSR

This is an important part of OLSR protocol and is generally derived from the shortest path algorithm as expressed under Dijkstra’s algorithm. RFC 3626 simplifies and adapts the algorithm for OLSR routing approach. It particularly incorporates the two-neighbourhood approach into the process. This makes it very efficient in WMN routing and the basis of the implementation of this study relies on this process.

It should be noted that this is the OLSR adaptation of description and steps in Dijkstra’s algorithm described in sections 3.2.2 & 3.2.3. Hence, the process can be highlighted in sequence as follows.

1. Add all registered symmetry on-hop neighbours to the routing table with a hop-count of 1.

2. For every symmetric one-hop neighbour, add all corresponding two-hop neighbours registered on that neighbour that have:
   
   (a) A symmetric link to the neighbour.
   (b) Not been added to the routing table already.

3. For every added node N in the routing table with hop-count n=2, entries from the TC set must be added to them provided that:

   (a) The originator in the TC entry == N
   (b) The destination has not already been added to the routing table.

Therefore, new entries are added with a hop-count of n+1 and next-hop as the next-hop registered on N’s routing entry.

1. Then, increment n with one and repeat step 3 over again until there are no entries in the routing table with hop-count == n+1

2. The MID set is queried for address aliases for all entries E in the routing table. If any aliases exist, then an entry is added to the routing table with hop-count set to E’s hop-count, and the next-hop set to E’s next-hop for every alias address.
D.6 Multi-point relaying

This concept was mentioned in chapter two and it is an optimised approach by OLSR to flood traffic in its domain. It is an attempt to minimise the number of unnecessary retransmission in the process. RFC 3626 suggests this as an optimised approach necessary for efficient routing process and not absolutely the best solution in finding the multi-point relay nodes. This may be open to optimisation research in the future.

The classical approach to flooding incorporates that all nodes retransmit received packets. To prevent infinite loop as a result of this, a sequence number is incorporated in the packet to ensure that packet is transmitted once. Therefore, if a node receives a packet with a sequence number equal or lower to the last registered retransmitted packet from the sender, the packet will be automatically dropped. Nodes in wireless multi-hop networks retransmit packets on the same interface that the packets arrived. This means that every re-transmitter will receive a duplicate packet from every symmetric neighbour that re-transmits the packet. In classical flooding approach in wireless multi-hop networks, the number of retransmission is n-1 where n is the number of nodes. Every transmitting node also receives the same packet. This process clearly requires some optimisation approach.

Routing in wired networks has incorporated some techniques whereby nodes are prevented from receiving packets they have already transmitted and this technique reduces the number of unnecessary retransmitted packets. OLSR uses the multi-point relaying approach to reduce the number of duplicate retransmissions while forwarding a broadcast packet. This approach restricts the retransmission of packets to a subset of nodes called multi-point relays. The number of MPR in a network domain depends on the size of the network. Every node will have to calculate its own set of MPRs as a subset of its symmetric neighbour nodes carefully chosen so that all 2-hop neighbours can be reached through a MPR.

Therefore, RFC 3626 proposes a simple heuristic algorithm for the selection of MPRs in a wireless networks. This states that for every node n in the network that can be reached from the local node by a minimum two symmetric hops, there exist a MPR m so that n has a symmetric link to m and m is a symmetric neighbour of the local node. This can be illustrated as follows in the Figure D.2. Node A selects the black nodes as MPRs so that all two hop nodes can be reached through MPR therefore, node B will not retransmit traffic from A that is to be flooded.

It should be noted that MPR optimisation is a key feature of OLSR which makes it efficient choice for WMN routing. This feature can be configured when deployed on wireless node through the configuration file. It also used to activate link quality in OLSR, this is possible when the nodes are configured to use link quality for MPR selection and routing.
D.7 Link state flooding

OLSR are further classified into three main modules and these are: neighbour sensing, multi-point relaying and link state flooding. Control traffic is generated based on the information retrieved from the set of repositories maintained by OLSR. These repositories are likewise updated dynamically by feedbacks received from the control messages. Figure D.3 adapted from Tønnesen (2004) shows the overview of OLSR repositories. This highlights how the link state information is flooded in OLSR for the purpose of route calculation.

![Diagram of MPR selection process in OLSR](image)
D.8 Cross-Layer design approach in OLSR

Figure D.3: Overview of OLSR information repositories

D.8 Cross-Layer design approach in OLSR

Figure D.4 further clarifies the operation of OLSR and the inter-operability with the plug-in program. The basic components of OLSR daemon which include socket parser, packet parser, information repositories tables and the scheduler are interacting with the plug-in program and its local tables to modify the operation of the OLSR protocol. The classical OLSR protocol determines its metric by using shortest path or Dijkstra’s algorithm. This research proposed to change the parameter and the computation of the metric and hence use this in the larger OLSR protocol. Figure D.4 shows the cross-layer approach in OLSR. This highlights the interaction of the plug-in program with the olsrd application.
Figure D.4: Cross-Layer approach in OLSR, adapted from Tønnesen (2004)